```
(.venv_clean) root@kind-name-wilts-fin-01:~/abstract-agent-runner#
python cli.py polyglot pov  agent-maxi-2.py http://135.181.71.13:8000 --
log-docker-to-stdout --verbose --timeout 1200
 INFO: [POLYGLOT] Loaded 33 problems from datasets/polyglot/
polyglot.json
DEBUG: [POLYGLOT]     Problem affine-cipher verified successfully (found
16 associated tests)
DEBUG: [POLYGLOT]     Problem beer-song verified successfully (found 8
associated tests)
DEBUG: [POLYGLOT]     Problem book-store verified successfully (found 20
associated tests)
DEBUG: [POLYGLOT]     Problem bottle-song verified successfully (found 7
associated tests)
DEBUG: [POLYGLOT]     Problem bowling verified successfully (found 31
associated tests)
DEBUG: [POLYGLOT]     Problem connect verified successfully (found 10
associated tests)
DEBUG: [POLYGLOT]     Problem dominoes verified successfully (found 13
associated tests)
DEBUG: [POLYGLOT]     Problem dot-dsl verified successfully (found 12
associated tests)
DEBUG: [POLYGLOT]     Problem food-chain verified successfully (found 10
associated tests)
DEBUG: [POLYGLOT]     Problem forth verified successfully (found 54
associated tests)
DEBUG: [POLYGLOT]     Problem go-counting verified successfully (found
11 associated tests)
DEBUG: [POLYGLOT]     Problem grade-school verified successfully (found
20 associated tests)
DEBUG: [POLYGLOT]     Problem grep verified successfully (found 25
associated tests)
DEBUG: [POLYGLOT]     Problem hangman verified successfully (found 7
associated tests)
DEBUG: [POLYGLOT]     Problem list-ops verified successfully (found 24
associated tests)
DEBUG: [POLYGLOT]     Problem phone-number verified successfully (found
21 associated tests)
DEBUG: [POLYGLOT]     Problem pig-latin verified successfully (found 22
associated tests)
DEBUG: [POLYGLOT]     Problem poker verified successfully (found 37
associated tests)
DEBUG: [POLYGLOT]     Problem pov verified successfully (found 15
associated tests)
DEBUG: [POLYGLOT]     Problem proverb verified successfully (found 8
associated tests)
DEBUG: [POLYGLOT]     Problem react verified successfully (found 14
associated tests)
DEBUG: [POLYGLOT]     Problem rest-api verified successfully (found 9
associated tests)
DEBUG: [POLYGLOT]     Problem robot-name verified successfully (found 4
associated tests)
DEBUG: [POLYGLOT]     Problem scale-generator verified successfully
(found 17 associated tests)
DEBUG: [POLYGLOT]     Problem sgf-parsing verified successfully (found
23 associated tests)
```

```
DEBUG: [POLYGLOT]     Problem simple-linked-list verified successfully
(found 20 associated tests)
DEBUG: [POLYGLOT]     Problem transpose verified successfully (found 12
associated tests)
DEBUG: [POLYGLOT]     Problem tree-building verified successfully (found
13 associated tests)
DEBUG: [POLYGLOT]     Problem two-bucket verified successfully (found 9
associated tests)
DEBUG: [POLYGLOT]     Problem variable-length-quantity verified
successfully (found 26 associated tests)
DEBUG: [POLYGLOT]     Problem wordy verified successfully (found 25
associated tests)
DEBUG: [POLYGLOT]     Problem zebra-puzzle verified successfully (found
2 associated tests)
DEBUG: [POLYGLOT]     Problem zipper verified successfully (found 14
associated tests)
 INFO: [POLYGLOT] Successfully loaded 33 problems
 INFO: Problem pov has 15 tests
 INFO: [SANDBOX] Checking gateway URL: http://135.181.71.13:8000
 INFO: [SANDBOX] Gateway URL http://135.181.71.13:8000 is valid
DEBUG: [SANDBOX] Stopping and deleting all containers
DEBUG: [SANDBOX] Stopped and deleted all containers
 INFO: [SANDBOX] Building Docker image: sandbox-image
[+] Building 0.3s (11/11) FINISHED
docker:default
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 1.04kB
0.0s
 => [internal] load metadata for docker.io/library/python:3.11-slim
0.2s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [1/6] FROM docker.io/library/python:3.11-
slim@sha256:b6000fc45f769f42c4c717dab2675bbb0ec6531c32a0483a2f78de0b7023
e71b
0.0s
 => => resolve docker.io/library/python:3.11-
slim@sha256:b6000fc45f769f42c4c717dab2675bbb0ec6531c32a0483a2f78de0b7023
e71b
0.0s
 => [internal] load build context
0.0s
 => => transferring context: 46B
0.0s
 => CACHED [2/6] RUN apt-get update &&     apt-get install -y --no-
install-recommends git patch diffutils &&     rm -rf /var/lib/apt/lists/
*
0.0s
 => CACHED [3/6] COPY sandbox_requirements.txt /tmp/
sandbox_requirements.txt
0.0s
```

```
 => CACHED [4/6] RUN pip install --no-cache-dir --upgrade pip &&     pip
install --no-cache-dir -r /tmp/sandbox_requirements.txt &&     rm /tmp/
sandbox_requirements.txt                                            0.0s
 => CACHED [5/6] RUN pip cache purge &&     rm -rf /root/.cache/pip
0.0s
 => CACHED [6/6] WORKDIR /sandbox
0.0s
 => exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:6a0d37b7a80c209efc720b51333a5af41822e44e249d5749ba8c3dcb2762dbcd
0.0s
 => => naming to docker.io/library/sandbox-image
0.0s
 INFO: [SANDBOX] Successfully built Docker image: sandbox-image
DEBUG: [SANDBOX] Found sandbox network: sandbox-network
 INFO: [SANDBOX] Building Docker image: sandbox-proxy-image
[+] Building 0.3s (8/8) FINISHED
docker:default
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 289B
0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine
0.2s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [1/3] FROM docker.io/library/
nginx:alpine@sha256:61e01287e546aac28a3f56839c136b31f590273f3b41187a36f4
6f6a03bbfe22
0.0s
 => [internal] load build context
0.0s
 => => transferring context: 41B
0.0s
 => CACHED [2/3] RUN apk add --no-cache gettext
0.0s
 => CACHED [3/3] COPY nginx.conf.template /tmp/nginx.conf.template
0.0s
 => exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:b38edefd62e5475e622d31caf7dabe385f28d175b050c8ae714b02c9758742a3
0.0s
 => => naming to docker.io/library/sandbox-proxy-image
0.0s
 INFO: [SANDBOX] Successfully built Docker image: sandbox-proxy-image
 INFO: [SANDBOX] Running sandbox proxy
DEBUG: [SANDBOX] Connected sandbox proxy to bridge network
DEBUG: [SANDBOX] Starting watchdog thread
```

```
DEBUG: [SANDBOX] Started watchdog thread
 INFO: [PROBLEM_SUITE] Starting sandbox to run agent for problem pov
DEBUG: [SANDBOX] Created sandbox temp directory for
<sandbox_tmpf8qgj5x7>: /tmp/tmpf8qgj5x7
DEBUG: [POLYGLOT] Copied main.py to /tmp/tmpf8qgj5x7/repo for pov
DEBUG: [POLYGLOT] Initializing git repository in /tmp/tmpf8qgj5x7/repo
for pov
DEBUG: [GIT] Initializing git repository in /tmp/tmpf8qgj5x7/repo
DEBUG: [GIT] Initialized git repository in /tmp/tmpf8qgj5x7/repo
DEBUG: [GIT] Adding all files in /tmp/tmpf8qgj5x7/repo
DEBUG: [GIT] Added all files in /tmp/tmpf8qgj5x7/repo
DEBUG: [GIT] Making initial commit: Initial commit
DEBUG: [GIT] Made initial commit: Initial commit
DEBUG: [POLYGLOT] Initialized git repository in /tmp/tmpf8qgj5x7/repo
for pov
DEBUG: [SANDBOX] Copied main Python script (/root/abstract-agent-runner/
problem_suites/AGENT_RUNNER.py) for <sandbox_tmpf8qgj5x7>: /tmp/
tmpf8qgj5x7/AGENT_RUNNER.py
DEBUG: [SANDBOX] Written input.json for <sandbox_tmpf8qgj5x7>: /tmp/
tmpf8qgj5x7/input.json
DEBUG: [SANDBOX] Running sandbox <sandbox_tmpf8qgj5x7>
DEBUG: [SANDBOX] Started sandbox runner thread for <sandbox_tmpf8qgj5x7>
DEBUG: [PROBLEM_SUITE] Started sandbox to run agent for problem pov
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Entered main()
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Reading input.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Read input.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Loading /sandbox/
agent.py
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Loaded /sandbox/
agent.py
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] agent_main() function
found in /sandbox/agent.py
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Entering agent's
agent_main()
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 🚀 AGENT STARTING
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:27:49,885 - agent -
INFO - [AGENT-START] Run ID: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:27:49,885 - agent -
INFO - [AGENT-START] Repo directory: /sandbox/repo
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:27:49,886 - agent -
INFO - [AGENT-START] Test mode: False
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:27:49,886 - agent -
INFO - [AGENT-START] Timeout: 1800s, Max steps: 400
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [DEBUG] Starting git initialization
check...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [DEBUG] Work directory: /sandbox/
repo
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [DEBUG] Before chdir - pwd shows: /sandbox/repo
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [DEBUG] After chdir - pwd shows: /sandbox/repo
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [DEBUG] Git repository already exists
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:27:49,894 - agent - INFO - [AGENT-START] Problem statement length: 2577 chars
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ────────────────────────────────────────────────────────────────────────────────
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 🔍 DETERMINING PROBLEM TYPE...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ────────────────────────────────────────────────────────────────────────────────
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:27:49,894 - agent - DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model: Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 436/219340 tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,776 - agent - INFO - [PROBLEM-TYPE] Determined: CREATE
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ✅ Problem type: CREATE
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ────────────────────────────────────────────────────────────────────────────────
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 🆕 CREATE TASK DETECTED - STARTING CREATE WORKFLOW
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ────────────────────────────────────────────────────────────────────────────────
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,777 - agent - INFO - [WORKFLOW] Starting CREATE task workflow
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ┌──────────────────────────────────────────────────────────────────────────────┐
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] │ 🆕 CREATE TASK WORKFLOW STARTING                                              │
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] └──────────────────────────────────────────────────────────────────────────────┘
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,777 - agent - INFO - [CREATE] Starting CREATE task workflow
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,777 - agent - INFO - [CREATE] Problem statement length: 2577 chars
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,777 - agent - INFO - [CREATE] Step 1: Generating code skeleton
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,778 - agent - INFO - [CREATE] Code skeleton generated: 636 chars
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,778 - agent - INFO - [CREATE] Step 2: Generating initial solution
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,778 - agent -
INFO - Starting multi-step reasoning solution generation
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:10,778 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 919/219340
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:16,669 - agent -
INFO - Multi-step reasoning solution generation completed successfully
with infinite loop validation
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:16,669 - agent -
INFO - Generated initial solution successfully using multi-step
reasoning
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:16,669 - agent -
INFO - [CREATE] Initial solution generated: 3965 chars
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:16,669 - agent -
INFO - [CREATE] Step 3: Extracting and writing solution files
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Created file: ./main.py
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:16,670 - agent -
INFO - [CREATE] Created 1 solution files
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ✅ Created 1 solution files
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:16,670 - agent -
INFO - [CREATE] Step 4: Generating test files
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:16,670 - agent -
INFO - Starting test cases generation
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:16,670 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 633/219340
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:23,620 - agent -
INFO - Step 1 - Testcase Generation completed
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:23,620 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 905/219340
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - Step 2 - Testcase check completed
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - Multi-step reasoning solution generation completed successfully
with infinite loop validation
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - Generated testcases successfully using multi-step reasoning
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [CREATE] Test cases generated: 4301 chars
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [CREATE] Step 5: Extracting and writing test files
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Created file: ./test_tree_test.py
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [CREATE] Created 1 test files
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ✅ Created 1 test files
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [CREATE] Time spent: 19.7s, Remaining timeout: 1720.3s
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [CREATE] Step 6: Starting iterative FIX workflow for refinement
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 🔧 Starting iterative refinement
workflow...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
┌───────────────────────────────────────────────────────────────────────
───────┐
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] │ 🔧 ITERATIVE FIX WORKFLOW STARTING
│
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
└───────────────────────────────────────────────────────────────────────
───────┘
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [ITERATIVE] Starting iterative FIX workflow
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [ITERATIVE] Run ID: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [ITERATIVE] Max steps: 30, Timeout: 1720.2525854110718s
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [ITERATIVE] Test runner: unittest (mode: FILE)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - State file not found at .agent_state.json. Starting with a fresh
state.
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,524 - agent -
INFO - [ITERATIVE] Loaded COT state: 0 existing thoughts
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,526 - agent -
INFO - [ITERATIVE] Initialized tool manager with 13 tools
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,527 - agent -
INFO - [ITERATIVE] Workflow started at 19:28:30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 🚀 Starting iterative workflow (max
30 steps, 1720.2525854110718s timeout)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,527 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 1/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 2629/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:30,527 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 2629/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:36,956 - agent -
INFO - LLM proposed action: search_in_all_files_content with args:
{'search_term': 'class Tree', 'case_sensitive': True}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:36,957 - agent -
INFO - Tool observation (first 300 chars): ./test_tree_test.py:6:class
TreeTest(unittest.TestCase):
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ./main.py:4:class Tree:...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:36,957 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 2/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 2704/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:36,958 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 2704/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:40,949 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./main.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:40,950 - agent -
INFO - Tool observation (first 300 chars): from json import dumps
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] class Tree:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def __init__(self, label,
children=None):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         self.label = label
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         self.children = children if
children is not None else []
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def __dict__(self):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         return {self.label:
[c.__dict__() for c in sorted(self.children)]}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def __str__(self, in...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:40,950 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 3/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3122/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:40,950 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3122/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:43,705 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./test_tree_test.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:43,705 - agent -
INFO - Tool observation (first 300 chars): import unittest
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] from main import Tree
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] class TreeTest(unittest.TestCase):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def test_empty_tree(self):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         tree = Tree('a')
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         expected = Tree('a')
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
self.assertEqual(tree.from_pov('a'), expected)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def test_single_node_tree(self):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         tree = Tree('a')
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         expected = Tr...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:43,705 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 4/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3497/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:43,706 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3497/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:48,322 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_tree_test.py']}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:48,322 - agent -
INFO - Running test command: python -m unittest test_tree_test
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:48,431 - agent -
INFO - Tool observation (first 300 chars): .....FF..FF........
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
======================================================================
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] FAIL: test_multi_node_tree_5
(test_tree_test.TreeTest.test_multi_node_tree_5)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Traceback (most recent call last):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]   File "/sandbox/repo/tes...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:48,431 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 5/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3666/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:48,431 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3666/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:55,704 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./test_tree_test.py', 'search_start_line': 35, 'search_end_line': 50}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:55,704 - agent -
INFO - Tool observation (first 300 chars): Lines 35-50 of ./
test_tree_test.py:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
self.assertEqual(tree.from_pov('b'), expected)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def
test_multi_node_tree_5(self):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         tree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         expected = Tree('d',
[Tree('b', [Tree('a', [Tree('e', [Tree('f'),...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:55,705 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 6/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3877/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:55,705 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3877/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:58,339 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./test_tree_test.py', 'search_start_line': 95, 'search_end_line': 105}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:58,339 - agent -
INFO - Tool observation (first 300 chars): Lines 95-105 of ./
test_tree_test.py:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         tree = Tree('a', [Tree('b'),
Tree('c')])
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         with
self.assertRaises(ValueError) as context:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]             tree.path_to('d', 'a')
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
self.assertEqual(str(context.exception), "No path found")
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def
test_path_between_unconnected_nodes(self):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]       ...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:58,340 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:58,340 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 7/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3961/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:28:58,340 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3961/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:06,727 - agent -
INFO - LLM proposed action: run_code with args: {'content': "from main
import Tree\n\n# Test the failing case 5\ntree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])\nexpected =
Tree('d', [Tree('b', [Tree('a', [Tree('e', [Tree('f'), Tree('g')]),
Tree('c')])])])\nresult = tree.from_pov('d')\n\nprint('Expected:',
expected)\nprint('Actual:', result)\nprint('Equal:', expected == result)
\n\n# Test the path issue\ntree2 = Tree('a', [Tree('b'), Tree('c',
[Tree('d')])])\ntry:\n    path = tree2.path_to('b', 'd')\n
print('Path found:', path)\nexcept ValueError as e:\n    print('Error:',
str(e))", 'file_path': './debug_test.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:06,755 - agent -
INFO - Tool observation (first 300 chars): Expected: {"d": [{"b": [{"a":
[{"c": []}, {"e": [{"f": []}, {"g": []}]}]}]}]}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Actual: {"d": [{"b": [{"a": [{"e":
[{"f": []}, {"g": []}]}]}, {"c": []}]}]}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Equal: False
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Path found: ['b', 'a', 'c', 'd']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:06,755 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 8/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3801/107929
tokens
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:06,755 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3801/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:13,491 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./main.py', 'search_start_line': 1, 'search_end_line': 50}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:13,492 - agent -
INFO - Tool observation (first 300 chars): Lines 1-50 of ./main.py:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] from json import dumps
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] class Tree:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def __init__(self, label,
children=None):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         self.label = label
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         self.children = children if
children is not None else []
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def __dict__(self):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         return {self.label:
[c.__dict__() for c in sorted(self.children)]}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:13,492 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 9/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3781/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:13,492 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3781/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:18,419 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./main.py', 'search_start_line': 50, 'search_end_line': 100}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:18,419 - agent -
INFO - Tool observation (first 300 chars): Lines 50-100 of ./main.py:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         if not target_node:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]             raise ValueError("Tree
could not be reoriented")
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         # Build parent map for the
entire tree
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         parent_map = {}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         build_parent_map(root)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         # Now we need to rebuild the
tree with from_node as the new root
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]        ...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:18,419 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 10/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3877/107929
tokens
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:18,419 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3877/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:20,909 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./main.py', 'search_start_line': 100, 'search_end_line': 150}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:20,910 - agent -
INFO - Tool observation (first 300 chars): Lines 100-115 of ./main.py:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]             path.append(node.label)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]             if node.label == target:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]                 return path[:]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]             for child in
node.children:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]                 result =
find_path(child, target, path)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]                 if result:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]                     return resul...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:20,910 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 11/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3893/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:20,910 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 3893/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:44,656 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def from_pov(self, from_node):\n        #
First, let\'s find if the tree contains the from_node\n        def
find_node(node, target):\n            if node.label == target:\n
return node\n            for child in node.children:\n
result = find_node(child, target)\n                if result:\n
return result\n            return None\n\n        # Build parent map to
navigate upwards\n        def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Get the
root of the current tree\n        def get_root(node):\n
current = node\n            while True:\n                parent =
parent_map.get(current.label)\n                if parent is None:\n
return current\n                current = parent\n\n        # Check if
from_node exists in the tree\n        root = get_root(self) if
\'parent_map\' in locals() else self\n        target_node =
find_node(root, from_node)\n        if not target_node:\n
raise ValueError("Tree could not be reoriented")\n\n        # Build
parent map for the entire tree\n        parent_map = {}\n
build_parent_map(root)\n\n        # Now we need to rebuild the tree with
from_node as the new root\n        # We\'ll do this by creating a new
tree structure\n        visited = set()\n        \n        def
build_new_tree(current_label):\n            if current_label in visited:
\n                return None\n            visited.add(current_label)\n
\n            # Find the original node\n            original_node =
find_node(root, current_label)\n            new_children = []\n
\n            # Add original children (except the one we came from)\n
if original_node:\n                for child in original_node.children:
\n                    if child.label not in visited:\n
new_child = build_new_tree(child.label)\n                        if
new_child:\n                            new_children.append(new_child)\n
\n            # Add parent as a child (except the one we came from)\n
parent = parent_map.get(current_label)\n            if parent and
parent.label not in visited:\n                new_parent_child =
build_new_tree(parent.label)\n                if new_parent_child:\n
new_children.append(new_parent_child)\n                    \n
return Tree(current_label, new_children)\n        \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)', 'replace': '    def from_pov(self,
from_node):\n        # First, let\'s find if the tree contains the
from_node\n        def find_node(node, target):\n            if
node.label == target:\n                return node\n            for
child in node.children:\n                result = find_node(child,
target)\n                if result:\n                    return result\n
return None\n\n        # Build parent map to navigate upwards\n
def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Now we need to rebuild the tree with
from_node as the new root\n        # We\'ll do this by creating a new
tree structure\n        visited = set()\n        \n        def
build_new_tree(current_label, came_from=None):\n            if
current_label in visited:\n                return None\n
visited.add(current_label)\n            \n            # Find the
original node\n            original_node = find_node(self,
current_label)\n            new_children = []\n            \n
# Add original children (except the one we came from)\n            if
original_node:\n                for child in original_node.children:\n
if child.label != came_from:\n                        new_child =
build_new_tree(child.label, current_label)\n                        if
new_child:\n                            new_children.append(new_child)\n
\n            # Add parent as a child (except the one we came from)\n
parent = parent_map.get(current_label)\n            if parent and
parent.label != came_from:\n                new_parent_child =
build_new_tree(parent.label, current_label)\n                if
new_parent_child:\n
new_children.append(new_parent_child)\n                    \n
return Tree(current_label, new_children)\n        \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:44,656 - agent -
INFO - Tool observation (first 300 chars): Error: You must get approval
before applying edits. Call get_approval_for_solution tool first....
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:44,657 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:44,657 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 12/30
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 4520/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:44,658 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 4520/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:53,422 - agent -
INFO - LLM proposed action: get_approval_for_solution with args:
{'solutions': ["Solution 1: Fix the tree reparenting logic by adding a
'came_from' parameter to track the direction when building the new tree.
This prevents cycles and ensures proper parent-child relationships. For
the path finding issue, fix the error handling to check if both nodes
exist first and raise the correct error messages ('No path found'
instead of 'Tree could not be reoriented' when from_node doesn't
exist).", 'Solution 2: Completely rewrite the tree reparenting logic
using a breadth-first approach starting from the target node. Build the
new tree by expanding outward in all directions (both to children and
parent) while maintaining proper relationships. For path finding, add
validation to check if both nodes exist in the original tree before
attempting to find a path, and ensure the correct error messages are
raised in all cases.'], 'selected_solution': 1, 'reason_for_selection':
"Solution 1 is more targeted and maintains the existing structure while
fixing the specific issues. The 'came_from' parameter approach is a
clean way to prevent cycles and ensure the tree is built correctly. This
solution is less invasive and more likely to maintain backward
compatibility while addressing all the failing test cases."}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:53,423 - agent -
INFO - Solutions proposed. Selected index: 1. Reason: Solution 1 is more
targeted and maintains the existing structure while fixing the specific
issues. The 'came_from' parameter approach is a clean way to prevent
cycles and ensure the tree is built correctly. This solution is less
invasive and more likely to maintain backward compatibility while
addressing all the failing test cases.
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:53,423 - agent -
INFO - Tool observation (first 300 chars): Approved...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:53,423 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 13/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 4831/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:29:53,423 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 4831/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:17,582 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def from_pov(self, from_node):\n        #
First, let\'s find if the tree contains the from_node\n        def
find_node(node, target):\n            if node.label == target:\n
return node\n            for child in node.children:\n
result = find_node(child, target)\n                if result:\n
return result\n            return None\n\n        # Build parent map to
navigate upwards\n        def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Get the
```

```
root of the current tree\n         def get_root(node):\n
current = node\n             while True:\n                 parent =
parent_map.get(current.label)\n                if parent is None:\n
return current\n                current = parent\n\n        # Check if
from_node exists in the tree\n        root = get_root(self) if
\'parent_map\' in locals() else self\n        target_node =
find_node(root, from_node)\n        if not target_node:\n
raise ValueError("Tree could not be reoriented")\n\n        # Build
parent map for the entire tree\n        parent_map = {}\n
build_parent_map(root)\n\n        # Now we need to rebuild the tree with
from_node as the new root\n        # We\'ll do this by creating a new
tree structure\n        visited = set()\n        \n        def
build_new_tree(current_label):\n            if current_label in visited:
\n                return None\n            visited.add(current_label)\n
\n            # Find the original node\n            original_node =
find_node(root, current_label)\n            new_children = []\n
\n            # Add original children (except the one we came from)\n
if original_node:\n                for child in original_node.children:
\n                    if child.label not in visited:\n
new_child = build_new_tree(child.label)\n                        if
new_child:\n                            new_children.append(new_child)\n
\n            # Add parent as a child (except the one we came from)\n
parent = parent_map.get(current_label)\n            if parent and
parent.label not in visited:\n                new_parent_child =
build_new_tree(parent.label)\n                if new_parent_child:\n
new_children.append(new_parent_child)\n                    \n
return Tree(current_label, new_children)\n        \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)', 'replace': '    def from_pov(self,
from_node):\n        # First, let\'s find if the tree contains the
from_node\n        def find_node(node, target):\n            if
node.label == target:\n                return node\n            for
child in node.children:\n                result = find_node(child,
target)\n                if result:\n                    return result\n
return None\n\n        # Build parent map to navigate upwards\n
def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Now we need to rebuild the tree with
from_node as the new root\n        # We\'ll do this by creating a new
tree structure\n        visited = set()\n        \n        def
build_new_tree(current_label, came_from=None):\n            if
current_label in visited:\n                return None\n
visited.add(current_label)\n            \n            # Find the
original node\n            original_node = find_node(self,
current_label)\n            new_children = []\n            \n
# Add original children (except the one we came from)\n            if
original_node:\n                for child in original_node.children:\n
if child.label != came_from:\n                        new_child =
build_new_tree(child.label, current_label)\n                        if
new_child:\n                            new_children.append(new_child)\n
\n            # Add parent as a child (except the one we came from)\n
```

```
parent = parent_map.get(current_label)\n             if parent and
parent.label != came_from:\n                 new_parent_child =
build_new_tree(parent.label, current_label)\n                 if
new_parent_child:\n
new_children.append(new_parent_child)\n                     \n
return Tree(current_label, new_children)\n         \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:17,583 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:17,583 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 14/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5194/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:17,584 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5194/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:30,602 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def path_to(self, from_node, to_node):\n
# Get tree from from_node\'s perspective\n        try:\n
new_tree = self.from_pov(from_node)\n        except ValueError:\n
raise ValueError("Tree could not be reoriented")\n        \n        #
Now find path in the new tree\n        def find_path(node, target,
path):\n            path.append(node.label)\n            if node.label
== target:\n                return path[:]\n            \n
for child in node.children:\n                result = find_path(child,
target, path)\n                if result:\n                    return
result\n                    \n            path.pop()\n            return
None\n            \n        path = find_path(new_tree, to_node, [])\n
if path is None:\n            raise ValueError("No path found")\n
return path', 'replace': '    def path_to(self, from_node, to_node):\n
# First check if both nodes exist in the tree\n        def
find_node(node, target):\n            if node.label == target:\n
return node\n            for child in node.children:\n
result = find_node(child, target)\n                if result:\n
return result\n            return None\n        \n        from_exists =
find_node(self, from_node) is not None\n        to_exists =
find_node(self, to_node) is not None\n        \n        if not
from_exists or not to_exists:\n            raise ValueError("No path
found")\n        \n        # Get tree from from_node\'s perspective\n
try:\n            new_tree = self.from_pov(from_node)\n        except
ValueError:\n            raise ValueError("No path found")\n        \n
# Now find path in the new tree\n        def find_path(node, target,
path):\n            path.append(node.label)\n            if node.label
== target:\n                return path[:]\n            \n
for child in node.children:\n                result = find_path(child,
target, path)\n                if result:\n                    return
result\n                    \n            path.pop()\n            return
None\n            \n        path = find_path(new_tree, to_node, [])\n
if path is None:\n            raise ValueError("No path found")\n
return path'}
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:30,604 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:30,604 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 15/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5294/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:30,605 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5294/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:33,504 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_tree_test.py']}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:33,505 - agent -
INFO - Running test command: python -m unittest test_tree_test
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:33,565 - agent -
INFO - Tool observation (first 300 chars): .....FF..F.........
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
======================================================================
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] FAIL: test_multi_node_tree_5
(test_tree_test.TreeTest.test_multi_node_tree_5)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Traceback (most recent call last):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]   File "/sandbox/repo/tes...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:33,565 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 16/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5389/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:33,566 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5389/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:37,688 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./test_tree_test.py', 'search_start_line': 35, 'search_end_line': 50}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:37,688 - agent -
INFO - Tool observation (first 300 chars): Lines 35-50 of ./
test_tree_test.py:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
self.assertEqual(tree.from_pov('b'), expected)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     def
test_multi_node_tree_5(self):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         tree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]         expected = Tree('d',
[Tree('b', [Tree('a', [Tree('e', [Tree('f'),...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:37,690 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:37,690 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 17/30
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5559/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:37,691 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5559/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:51,859 - agent -
INFO - LLM proposed action: run_code with args: {'content': "from main
import Tree\n\n# Test case 5: from_pov('d')\ntree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])\nexpected =
Tree('d', [Tree('b', [Tree('a', [Tree('e', [Tree('f'), Tree('g')]),
Tree('c')])])])\nresult = tree.from_pov('d')\n\nprint('=== Test Case 5
===')\nprint('Expected structure:')\nprint(f'Root: {expected.label}')
\nprint(f'Children: {[child.label for child in expected.children]}')\nif
expected.children:\n    print(f'Child 0 children: {[child.label for
child in expected.children[0].children]}')\n    if
expected.children[0].children:\n        print(f'Grandchild 0 children:
{[child.label for child in expected.children[0].children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result.label}')
\nprint(f'Children: {[child.label for child in result.children]}')\nif
result.children:\n    print(f'Child 0 children: {[child.label for child
in result.children[0].children]}')\n    if result.children[0].children:
\n        print(f'Grandchild 0 children: {[child.label for child in
result.children[0].children[0].children]}')\n\nprint('\\nEqual:',
expected == result)\n\n# Test case 6: from_pov('e')\ntree2 = Tree('a',
[Tree('b', [Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])
\nexpected2 = Tree('e', [Tree('a', [Tree('b', [Tree('c'), Tree('d')]),
Tree('f'), Tree('g')])])\nresult2 = tree2.from_pov('e')\n\nprint('\\n===
Test Case 6 ===')\nprint('Expected structure:')\nprint(f'Root:
{expected2.label}')\nprint(f'Children: {[child.label for child in
expected2.children]}')\nif expected2.children:\n    print(f'Child 0
children: {[child.label for child in expected2.children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result2.label}')
\nprint(f'Children: {[child.label for child in result2.children]}')\nif
result2.children:\n    print(f'Child 0 children: {[child.label for child
in result2.children[0].children]}')\n\nprint('\\nEqual:', expected2 ==
result2)\n\n# Test path between unconnected nodes\ntree3 = Tree('a',
[Tree('b'), Tree('c', [Tree('d')])])\ntry:\n    path =
tree3.path_to('b', 'd')\n    print('\\n=== Path Test ===')\n
print('Path found:', path)\nexcept ValueError as e:\n    print('\\n===
Path Test ===')\n    print('Error raised:', str(e))", 'file_path': './
debug_test.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:51,892 - agent -
INFO - Tool observation (first 300 chars): === Test Case 5 ===
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Expected structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: d
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['b']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: ['a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Grandchild 0 children: ['e', 'c']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Actual structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: d
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['b']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: ['c', 'a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Grandchild 0 children: []
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Equal: False
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] === Test Case 6 ===
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Expected structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: e
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] C...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:51,892 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 18/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5851/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:30:51,893 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 5851/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:16,595 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def from_pov(self, from_node):\n        #
First, let\'s find if the tree contains the from_node\n        def
find_node(node, target):\n            if node.label == target:\n
return node\n            for child in node.children:\n
result = find_node(child, target)\n                if result:\n
return result\n            return None\n\n        # Build parent map to
navigate upwards\n        def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Now we need to rebuild the tree with
from_node as the new root\n        # We\'ll do this by creating a new
tree structure\n        visited = set()\n        \n        def
build_new_tree(current_label, came_from=None):\n            if
current_label in visited:\n                return None\n
visited.add(current_label)\n            \n            # Find the
original node\n            original_node = find_node(self,
current_label)\n            new_children = []\n            \n
# Add original children (except the one we came from)\n            if
original_node:\n                for child in original_node.children:\n
if child.label != came_from:\n                        new_child =
build_new_tree(child.label, current_label)\n                        if
new_child:\n                            new_children.append(new_child)\n
\n            # Add parent as a child (except the one we came from)\n
parent = parent_map.get(current_label)\n            if parent and
parent.label != came_from:\n                new_parent_child =
build_new_tree(parent.label, current_label)\n                if
new_parent_child:\n
new_children.append(new_parent_child)\n                    \n
return Tree(current_label, new_children)\n        \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)', 'replace': '    def from_pov(self,
from_node):\n        # First, let\'s find if the tree contains the
from_node\n        def find_node(node, target):\n            if
```

```
node.label == target:\n                return node\n            for
child in node.children:\n                result = find_node(child,
target)\n                if result:\n                    return result\n
return None\n\n        # Build parent map to navigate upwards\n
def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Now we need to rebuild the tree with
from_node as the new root\n        # We\'ll do this by creating a new
tree structure\n        visited = set()\n        \n        def
build_new_tree(current_label, came_from=None):\n            if
current_label in visited:\n                return None\n
visited.add(current_label)\n            \n            # Find the
original node\n            original_node = find_node(self,
current_label)\n            new_children = []\n            \n
# Add original children (except the one we came from)\n            if
original_node:\n                for child in original_node.children:\n
if child.label != came_from:\n                        new_child =
build_new_tree(child.label, current_label)\n                        if
new_child:\n                            new_children.append(new_child)\n
\n            # Add parent as a child (except the one we came from)\n
parent = parent_map.get(current_label)\n            if parent and
parent.label != came_from:\n                new_parent_child =
build_new_tree(parent.label, current_label)\n                if
new_parent_child:\n
new_children.append(new_parent_child)\n                    \n
return Tree(current_label, new_children)\n        \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:16,596 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:16,596 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 19/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 6507/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:16,597 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 6507/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:39,894 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def from_pov(self, from_node):\n        #
First, let\'s find if the tree contains the from_node\n        def
find_node(node, target):\n            if node.label == target:\n
return node\n            for child in node.children:\n
result = find_node(child, target)\n                if result:\n
return result\n            return None\n\n        # Build parent map to
navigate upwards\n        def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
```

```
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Now we need to rebuild the tree with
from_node as the new root\n        # We\'ll do this by creating a new
tree structure\n        visited = set()\n        \n        def
build_new_tree(current_label, came_from=None):\n            if
current_label in visited:\n                return None\n
visited.add(current_label)\n            \n            # Find the
original node\n            original_node = find_node(self,
current_label)\n            new_children = []\n            \n
# Add original children (except the one we came from)\n            if
original_node:\n                for child in original_node.children:\n
if child.label != came_from:\n                        new_child =
build_new_tree(child.label, current_label)\n                        if
new_child:\n                            new_children.append(new_child)\n
\n            # Add parent as a child (except the one we came from)\n
parent = parent_map.get(current_label)\n            if parent and
parent.label != came_from:\n                new_parent_child =
build_new_tree(parent.label, current_label)\n                if
new_parent_child:\n
new_children.append(new_parent_child)\n            \n
return Tree(current_label, new_children)\n        \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)', 'replace': '    def from_pov(self,
from_node):\n        # First, let\'s find if the tree contains the
from_node\n        def find_node(node, target):\n            if
node.label == target:\n                return node\n            for
child in node.children:\n                result = find_node(child,
target)\n                if result:\n                    return result\n
return None\n\n        # Build parent map to navigate upwards\n
def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Build a map of all nodes and their
original children\n        def build_children_map(node):\n
children_map[node.label] = [child.label for child in node.children]\n
for child in node.children:\n                build_children_map(child)\n
\n        children_map = {}\n        build_children_map(self)\n\n
# Now we need to rebuild the tree with from_node as the new root\n
# We\'ll do this by creating a new tree structure\n        visited =
set()\n        \n        def build_new_tree(current_label,
came_from=None):\n            if current_label in visited:\n
return None\n            visited.add(current_label)\n            \n
new_children = []\n            \n            # Add original children
(except the one we came from)\n            for child_label in
children_map.get(current_label, []):\n                if child_label !=
came_from:\n                    new_child = build_new_tree(child_label,
current_label)\n                    if new_child:\n
new_children.append(new_child)\n            \n            # Add parent
as a child (except the one we came from)\n            parent =
parent_map.get(current_label)\n            if parent and parent.label !=
came_from:\n                new_parent_child =
build_new_tree(parent.label, current_label)\n                if
new_parent_child:\n
new_children.append(new_parent_child)\n                    \n
return Tree(current_label, new_children)\n        \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:39,896 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:39,896 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 20/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 7078/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:39,897 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 7078/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:52,925 - agent -
INFO - LLM proposed action: run_code with args: {'content': "from main
import Tree\n\n# Test case 5: from_pov('d')\ntree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])\nexpected =
Tree('d', [Tree('b', [Tree('a', [Tree('e', [Tree('f'), Tree('g')]),
Tree('c')])])])\nresult = tree.from_pov('d')\n\nprint('=== Test Case 5
===')\nprint('Expected structure:')\nprint(f'Root: {expected.label}')
\nprint(f'Children: {[child.label for child in expected.children]}')\nif
expected.children:\n    print(f'Child 0 children: {[child.label for
child in expected.children[0].children]}')\n    if
expected.children[0].children:\n        print(f'Grandchild 0 children:
{[child.label for child in expected.children[0].children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result.label}')
\nprint(f'Children: {[child.label for child in result.children]}')\nif
result.children:\n    print(f'Child 0 children: {[child.label for child
in result.children[0].children]}')\n    if result.children[0].children:
\n        print(f'Grandchild 0 children: {[child.label for child in
result.children[0].children[0].children]}')\n\nprint('\\nEqual:',
expected == result)\n\n# Test case 6: from_pov('e')\ntree2 = Tree('a',
[Tree('b', [Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])
\nexpected2 = Tree('e', [Tree('a', [Tree('b', [Tree('c'), Tree('d')]),
Tree('f'), Tree('g')])])\nresult2 = tree2.from_pov('e')\n\nprint('\\n===
Test Case 6 ===')\nprint('Expected structure:')\nprint(f'Root:
{expected2.label}')\nprint(f'Children: {[child.label for child in
expected2.children]}')\nif expected2.children:\n    print(f'Child 0
children: {[child.label for child in expected2.children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result2.label}')
\nprint(f'Children: {[child.label for child in result2.children]}')\nif
result2.children:\n    print(f'Child 0 children: {[child.label for child
in result2.children[0].children]}')\n\nprint('\\nEqual:', expected2 ==
result2)", 'file_path': './debug_test.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:52,958 - agent -
INFO - Tool observation (first 300 chars): === Test Case 5 ===
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Expected structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: d
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['b']
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: ['a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Grandchild 0 children: ['e', 'c']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Actual structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: d
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['b']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: ['c', 'a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Grandchild 0 children: []
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Equal: False
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] === Test Case 6 ===
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Expected structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: e
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] C...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:52,958 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 21/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 7238/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:31:52,959 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 7238/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:20,701 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def from_pov(self, from_node):\n        #
First, let\'s find if the tree contains the from_node\n        def
find_node(node, target):\n            if node.label == target:\n
return node\n            for child in node.children:\n
result = find_node(child, target)\n                if result:\n
return result\n            return None\n\n        # Build parent map to
navigate upwards\n        def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Build a map of all nodes and their
original children\n        def build_children_map(node):\n
children_map[node.label] = [child.label for child in node.children]\n
for child in node.children:\n                build_children_map(child)\n
\n        children_map = {}\n        build_children_map(self)\n\n
# Now we need to rebuild the tree with from_node as the new root\n
# We\'ll do this by creating a new tree structure\n        visited =
set()\n        \n        def build_new_tree(current_label,
came_from=None):\n            if current_label in visited:\n
return None\n            visited.add(current_label)\n            \n
new_children = []\n            \n            # Add original children
(except the one we came from)\n            for child_label in
children_map.get(current_label, []):\n                if child_label !=
came_from:\n                    new_child = build_new_tree(child_label,
current_label)\n                    if new_child:\n
new_children.append(new_child)\n            \n            # Add parent
```

```
as a child (except the one we came from)\n            parent =
parent_map.get(current_label)\n            if parent and parent.label !=
came_from:\n                new_parent_child =
build_new_tree(parent.label, current_label)\n                if
new_parent_child:\n
new_children.append(new_parent_child)\n                    \n
return Tree(current_label, new_children)\n        \n        # Reset
visited for the actual tree building\n        visited = set()\n
return build_new_tree(from_node)', 'replace': '    def from_pov(self,
from_node):\n        # First, let\'s find if the tree contains the
from_node\n        def find_node(node, target):\n            if
node.label == target:\n                return node\n            for
child in node.children:\n                result = find_node(child,
target)\n                if result:\n                    return result\n
return None\n\n        # Build parent map to navigate upwards\n
def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Build a map of all nodes and their
original children\n        def build_children_map(node):\n
children_map[node.label] = [child.label for child in node.children]\n
for child in node.children:\n                build_children_map(child)\n
\n        children_map = {}\n        build_children_map(self)\n\n
# Build the new tree structure by determining new parent-child
relationships\n        # In the new tree, each node\'s children are:\n
# 1. Its original children (except the one that leads back to the new
root)\n        # 2. Its original parent (except the one that leads back
to the new root)\n        \n        # First, find the path from the
original root to the new root\n        def find_path_to_root(node_label,
path):\n            path.append(node_label)\n            parent =
parent_map.get(node_label)\n            if parent is None:\n
return path\n            return find_path_to_root(parent.label, path)\n
\n        path_to_new_root = find_path_to_root(from_node, [])\n
path_to_new_root.reverse()  # Now it\'s from original root to new root\n
\n        # Build new children map for the reparented tree\n
new_children_map = {}\n        \n        # For each node, determine its
new children\n        for node_label in children_map:\n
new_children = []\n            \n            # Original children become
children, except the one that\'s on the path to new root\n
for child_label in children_map[node_label]:\n                if
child_label not in path_to_new_root or
path_to_new_root.index(child_label) !=
path_to_new_root.index(node_label) + 1:\n
new_children.append(child_label)\n            \n            # Original
parent becomes a child, except the one that\'s on the path to new root\n
parent = parent_map.get(node_label)\n            if parent:\n
if parent.label not in path_to_new_root or
path_to_new_root.index(parent.label) !=
path_to_new_root.index(node_label) - 1:\n
new_children.append(parent.label)\n            \n
new_children_map[node_label] = new_children\n        \n        # Now
build the actual tree structure\n        visited = set()\n        \n
```

```
def build_tree_recursive(node_label):\n            if node_label in
visited:\n                return None\n
visited.add(node_label)\n            \n            children = []\n
for child_label in new_children_map.get(node_label, []):\n
child_tree = build_tree_recursive(child_label)\n                if
child_tree:\n                    children.append(child_tree)\n
\n            return Tree(node_label, children)\n        \n
return build_tree_recursive(from_node)'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:20,703 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:20,705 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:20,705 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 22/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 7886/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:20,706 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 7886/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:34,905 - agent -
INFO - LLM proposed action: run_code with args: {'content': "from main
import Tree\n\n# Test case 5: from_pov('d')\ntree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])\nexpected =
Tree('d', [Tree('b', [Tree('a', [Tree('e', [Tree('f'), Tree('g')]),
Tree('c')])])])\nresult = tree.from_pov('d')\n\nprint('=== Test Case 5
===')\nprint('Expected structure:')\nprint(f'Root: {expected.label}')
\nprint(f'Children: {[child.label for child in expected.children]}')\nif
expected.children:\n    print(f'Child 0 children: {[child.label for
child in expected.children[0].children]}')\n    if
expected.children[0].children:\n        print(f'Grandchild 0 children:
{[child.label for child in expected.children[0].children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result.label}')
\nprint(f'Children: {[child.label for child in result.children]}')\nif
result.children:\n    print(f'Child 0 children: {[child.label for child
in result.children[0].children]}')\n    if result.children[0].children:
\n        print(f'Grandchild 0 children: {[child.label for child in
result.children[0].children[0].children]}')\n\nprint('\\nEqual:',
expected == result)\n\n# Test case 6: from_pov('e')\ntree2 = Tree('a',
[Tree('b', [Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])
\nexpected2 = Tree('e', [Tree('a', [Tree('b', [Tree('c'), Tree('d')]),
Tree('f'), Tree('g')])])\nresult2 = tree2.from_pov('e')\n\nprint('\\n===
Test Case 6 ===')\nprint('Expected structure:')\nprint(f'Root:
{expected2.label}')\nprint(f'Children: {[child.label for child in
expected2.children]}')\nif expected2.children:\n    print(f'Child 0
children: {[child.label for child in expected2.children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result2.label}')
\nprint(f'Children: {[child.label for child in result2.children]}')\nif
result2.children:\n    print(f'Child 0 children: {[child.label for child
in result2.children[0].children]}')\n\nprint('\\nEqual:', expected2 ==
result2)", 'file_path': './debug_test.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:34,944 - agent -
INFO - Tool observation (first 300 chars): Error running code: Traceback
(most recent call last):
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]   File "/sandbox/repo/./
debug_test.py", line 6, in <module>
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     result = tree.from_pov('d')
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]              ^^^^^^^^^^^^^^^^^^
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]   File "/sandbox/repo/main.py", line
87, in from_pov
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     if parent.label not in
path_to_new_root or path_to_new_root.inde...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:34,944 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 23/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8090/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:34,945 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8090/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:41,348 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': "            # Original parent becomes a child,
except the one that's on the path to new root\n            parent =
parent_map.get(node_label)\n            if parent:\n                if
parent.label not in path_to_new_root or
path_to_new_root.index(parent.label) !=
path_to_new_root.index(node_label) - 1:\n
new_children.append(parent.label)", 'replace': "            # Original
parent becomes a child, except the one that's on the path to new root\n
parent = parent_map.get(node_label)\n            if parent:\n
parent_in_path = parent.label in path_to_new_root\n                if
not parent_in_path or (parent_in_path and
path_to_new_root.index(parent.label) !=
path_to_new_root.index(node_label) - 1):\n
new_children.append(parent.label)"}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:41,349 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:41,349 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 24/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8244/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:41,351 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8244/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:55,705 - agent -
INFO - LLM proposed action: run_code with args: {'content': "from main
import Tree\n\n# Test case 5: from_pov('d')\ntree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])\nexpected =
Tree('d', [Tree('b', [Tree('a', [Tree('e', [Tree('f'), Tree('g')]),
Tree('c')])])])\nresult = tree.from_pov('d')\n\nprint('=== Test Case 5
===')\nprint('Expected structure:')\nprint(f'Root: {expected.label}')
\nprint(f'Children: {[child.label for child in expected.children]}')\nif
expected.children:\n    print(f'Child 0 children: {[child.label for
child in expected.children[0].children]}')\n    if
expected.children[0].children:\n        print(f'Grandchild 0 children:
```

```
{[child.label for child in expected.children[0].children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result.label}')
\nprint(f'Children: {[child.label for child in result.children]}')\nif
result.children:\n    print(f'Child 0 children: {[child.label for child
in result.children[0].children]}')\n    if result.children[0].children:
\n        print(f'Grandchild 0 children: {[child.label for child in
result.children[0].children[0].children]}')\n\nprint('\\nEqual:',
expected == result)\n\n# Test case 6: from_pov('e')\ntree2 = Tree('a',
[Tree('b', [Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])
\nexpected2 = Tree('e', [Tree('a', [Tree('b', [Tree('c'), Tree('d')]),
Tree('f'), Tree('g')])])\nresult2 = tree2.from_pov('e')\n\nprint('\\n===
Test Case 6 ===')\nprint('Expected structure:')\nprint(f'Root:
{expected2.label}')\nprint(f'Children: {[child.label for child in
expected2.children]}')\nif expected2.children:\n    print(f'Child 0
children: {[child.label for child in expected2.children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result2.label}')
\nprint(f'Children: {[child.label for child in result2.children]}')\nif
result2.children:\n    print(f'Child 0 children: {[child.label for child
in result2.children[0].children]}')\n\nprint('\\nEqual:', expected2 ==
result2)", 'file_path': './debug_test.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:55,737 - agent -
INFO - Tool observation (first 300 chars): Error running code: Traceback
(most recent call last):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]   File "/sandbox/repo/./
debug_test.py", line 6, in <module>
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     result = tree.from_pov('d')
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]              ^^^^^^^^^^^^^^^^^^
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]   File "/sandbox/repo/main.py", line
88, in from_pov
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]     if not parent_in_path or
(parent_in_path and path_to_new_root.in...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:55,737 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 25/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8487/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:32:55,739 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8487/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:02,749 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': "            # Original parent becomes a child,
except the one that's on the path to new root\n            parent =
parent_map.get(node_label)\n            if parent:\n
parent_in_path = parent.label in path_to_new_root\n                if
not parent_in_path or (parent_in_path and
path_to_new_root.index(parent.label) !=
path_to_new_root.index(node_label) - 1):\n
new_children.append(parent.label)", 'replace': "            # Original
parent becomes a child, except the one that's on the path to new root\n
parent = parent_map.get(node_label)\n            if parent:\n
parent_in_path = parent.label in path_to_new_root\n
node_in_path = node_label in path_to_new_root\n                \n
if not parent_in_path or not node_in_path or (parent_in_path and
node_in_path and path_to_new_root.index(parent.label) !=
path_to_new_root.index(node_label) - 1):\n
new_children.append(parent.label)"}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:02,751 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:02,751 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 26/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8599/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:02,752 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8599/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:21,622 - agent -
INFO - LLM proposed action: run_code with args: {'content': "from main
import Tree\n\n# Test case 5: from_pov('d')\ntree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])\nexpected =
Tree('d', [Tree('b', [Tree('a', [Tree('e', [Tree('f'), Tree('g')]),
Tree('c')])])])\nresult = tree.from_pov('d')\n\nprint('=== Test Case 5
===')\nprint('Expected structure:')\nprint(f'Root: {expected.label}')
\nprint(f'Children: {[child.label for child in expected.children]}')\nif
expected.children:\n    print(f'Child 0 children: {[child.label for
child in expected.children[0].children]}')\n    if
expected.children[0].children:\n        print(f'Grandchild 0 children:
{[child.label for child in expected.children[0].children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result.label}')
\nprint(f'Children: {[child.label for child in result.children]}')\nif
result.children:\n    print(f'Child 0 children: {[child.label for child
in result.children[0].children]}')\n    if result.children[0].children:
\n        print(f'Grandchild 0 children: {[child.label for child in
result.children[0].children[0].children]}')\n\nprint('\\nEqual:',
expected == result)\n\n# Test case 6: from_pov('e')\ntree2 = Tree('a',
[Tree('b', [Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])
\nexpected2 = Tree('e', [Tree('a', [Tree('b', [Tree('c'), Tree('d')]),
Tree('f'), Tree('g')])])\nresult2 = tree2.from_pov('e')\n\nprint('\\n===
Test Case 6 ===')\nprint('Expected structure:')\nprint(f'Root:
{expected2.label}')\nprint(f'Children: {[child.label for child in
expected2.children]}')\nif expected2.children:\n    print(f'Child 0
children: {[child.label for child in expected2.children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result2.label}')
\nprint(f'Children: {[child.label for child in result2.children]}')\nif
result2.children:\n    print(f'Child 0 children: {[child.label for child
in result2.children[0].children]}')\n\nprint('\\nEqual:', expected2 ==
result2)", 'file_path': './debug_test.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:21,655 - agent -
INFO - Tool observation (first 300 chars): === Test Case 5 ===
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Expected structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: d
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['b']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: ['a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Grandchild 0 children: ['e', 'c']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Actual structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: d
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: []
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Equal: False
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] === Test Case 6 ===
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Expected structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: e
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: ['b', 'f', 'g']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Actual structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: ...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:21,657 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:21,657 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 27/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8846/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:21,658 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 8846/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:56,804 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def from_pov(self, from_node):\n        #
First, let\'s find if the tree contains the from_node\n        def
find_node(node, target):\n            if node.label == target:\n
return node\n            for child in node.children:\n
result = find_node(child, target)\n                if result:\n
return result\n            return None\n\n        # Build parent map to
navigate upwards\n        def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Build a map of all nodes and their
original children\n        def build_children_map(node):\n
children_map[node.label] = [child.label for child in node.children]\n
for child in node.children:\n                build_children_map(child)\n
\n        children_map = {}\n        build_children_map(self)\n\n
# Build the new tree structure by determining new parent-child
relationships\n        # In the new tree, each node\'s children are:\n
# 1. Its original children (except the one that leads back to the new
root)\n        # 2. Its original parent (except the one that leads back
to the new root)\n        \n        # First, find the path from the
original root to the new root\n        def find_path_to_root(node_label,
path):\n            path.append(node_label)\n            parent =
parent_map.get(node_label)\n            if parent is None:\n
return path\n            return find_path_to_root(parent.label, path)\n
\n        path_to_new_root = find_path_to_root(from_node, [])\n
path_to_new_root.reverse()  # Now it\'s from original root to new root\n
\n        # Build new children map for the reparented tree\n
new_children_map = {}\n        \n        # For each node, determine its
new children\n        for node_label in children_map:\n
```

```
new_children = []\n            \n            # Original children become
children, except the one that\'s on the path to new root\n
for child_label in children_map[node_label]:\n                if
child_label not in path_to_new_root or
path_to_new_root.index(child_label) !=
path_to_new_root.index(node_label) + 1:\n
new_children.append(child_label)\n            \n            # Original
parent becomes a child, except the one that\'s on the path to new root\n
parent = parent_map.get(node_label)\n            if parent:\n
parent_in_path = parent.label in path_to_new_root\n
node_in_path = node_label in path_to_new_root\n                \n
if not parent_in_path or not node_in_path or (parent_in_path and
node_in_path and path_to_new_root.index(parent.label) !=
path_to_new_root.index(node_label) - 1):\n
new_children.append(parent.label)\n            \n
new_children_map[node_label] = new_children\n        \n        # Now
build the actual tree structure\n        visited = set()\n        \n
def build_tree_recursive(node_label):\n            if node_label in
visited:\n                return None\n
visited.add(node_label)\n            \n            children = []\n
for child_label in new_children_map.get(node_label, []):\n
child_tree = build_tree_recursive(child_label)\n                if
child_tree:\n                    children.append(child_tree)\n
\n            return Tree(node_label, children)\n        \n
return build_tree_recursive(from_node)', 'replace': '    def
from_pov(self, from_node):\n        # First, let\'s find if the tree
contains the from_node\n        def find_node(node, target):\n
if node.label == target:\n                return node\n            for
child in node.children:\n                result = find_node(child,
target)\n                if result:\n                    return result\n
return None\n\n        # Build parent map to navigate upwards\n
def build_parent_map(node, parent=None):\n
parent_map[node.label] = parent\n            for child in node.children:
\n                build_parent_map(child, node)\n\n        # Check if
from_node exists in the tree\n        target_node = find_node(self,
from_node)\n        if not target_node:\n            raise
ValueError("Tree could not be reoriented")\n\n        # Build parent map
for the entire tree\n        parent_map = {}\n
build_parent_map(self)\n\n        # Build a map of all nodes and their
original children\n        def build_children_map(node):\n
children_map[node.label] = [child.label for child in node.children]\n
for child in node.children:\n                build_children_map(child)\n
\n        children_map = {}\n        build_children_map(self)\n\n
# Build the new tree using a simpler approach\n        # Start from the
new root and recursively build the tree\n        visited = set()\n
\n        def build_tree_recursive(current_label, came_from=None):\n
if current_label in visited:\n                return None\n
visited.add(current_label)\n            \n            new_children = []
\n            \n            # Add original children (except the one we
came from)\n            for child_label in
children_map.get(current_label, []):\n                if child_label !=
came_from:\n                    child_tree =
build_tree_recursive(child_label, current_label)\n                    if
child_tree:\n                        new_children.append(child_tree)\n
\n            # Add parent as a child (except the one we came from)\n
parent = parent_map.get(current_label)\n            if parent and
```

```
parent.label != came_from:\n                parent_tree =
build_tree_recursive(parent.label, current_label)\n                if
parent_tree:\n                    new_children.append(parent_tree)\n
\n            return Tree(current_label, new_children)\n        \n
return build_tree_recursive(from_node)'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:56,805 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:56,805 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 28/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 9525/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:33:56,807 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 9525/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:20,186 - agent -
INFO - LLM proposed action: run_code with args: {'content': "from main
import Tree\n\n# Test case 5: from_pov('d')\ntree = Tree('a', [Tree('b',
[Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])\nexpected =
Tree('d', [Tree('b', [Tree('a', [Tree('e', [Tree('f'), Tree('g')]),
Tree('c')])])])\nresult = tree.from_pov('d')\n\nprint('=== Test Case 5
===')\nprint('Expected structure:')\nprint(f'Root: {expected.label}')
\nprint(f'Children: {[child.label for child in expected.children]}')\nif
expected.children:\n    print(f'Child 0 children: {[child.label for
child in expected.children[0].children]}')\n    if
expected.children[0].children:\n        print(f'Grandchild 0 children:
{[child.label for child in expected.children[0].children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result.label}')
\nprint(f'Children: {[child.label for child in result.children]}')\nif
result.children:\n    print(f'Child 0 children: {[child.label for child
in result.children[0].children]}')\n    if result.children[0].children:
\n        print(f'Grandchild 0 children: {[child.label for child in
result.children[0].children[0].children]}')\n\nprint('\\nEqual:',
expected == result)\n\n# Test case 6: from_pov('e')\ntree2 = Tree('a',
[Tree('b', [Tree('c'), Tree('d')]), Tree('e', [Tree('f'), Tree('g')])])
\nexpected2 = Tree('e', [Tree('a', [Tree('b', [Tree('c'), Tree('d')]),
Tree('f'), Tree('g')])])\nresult2 = tree2.from_pov('e')\n\nprint('\\n===
Test Case 6 ===')\nprint('Expected structure:')\nprint(f'Root:
{expected2.label}')\nprint(f'Children: {[child.label for child in
expected2.children]}')\nif expected2.children:\n    print(f'Child 0
children: {[child.label for child in expected2.children[0].children]}')
\n\nprint('\\nActual structure:')\nprint(f'Root: {result2.label}')
\nprint(f'Children: {[child.label for child in result2.children]}')\nif
result2.children:\n    print(f'Child 0 children: {[child.label for child
in result2.children[0].children]}')\n\nprint('\\nEqual:', expected2 ==
result2)", 'file_path': './debug_test.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:20,218 - agent -
INFO - Tool observation (first 300 chars): === Test Case 5 ===
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Expected structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: d
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['b']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: ['a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Grandchild 0 children: ['e', 'c']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Actual structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: d
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['b']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: ['c', 'a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Grandchild 0 children: []
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Equal: False
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] === Test Case 6 ===
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Expected structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: e
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['a']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] C...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:20,218 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 29/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 9792/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:20,219 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 9792/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:25,771 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_tree_test.py']}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:25,771 - agent -
INFO - Running test command: python -m unittest test_tree_test
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:25,830 - agent -
INFO - Tool observation (first 300 chars): .....FF..F.........
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
======================================================================
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] FAIL: test_multi_node_tree_5
(test_tree_test.TreeTest.test_multi_node_tree_5)
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Traceback (most recent call last):
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]   File "/sandbox/repo/tes...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:25,830 - agent -
INFO - [RUN:199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b] Step 30/30
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 9970/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:25,832 - agent -
DEBUG - [REQUEST] run_id: 199ea0dc-0837-4bc8-b8ab-1c58fdcc5a2b, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [CONTEXT] Messages fit: 9970/107929
tokens
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:40,186 - agent -
INFO - LLM proposed action: run_code with args: {'content': "from main
import Tree\n\n# Debug the path finding issue\ntree = Tree('a',
[Tree('b'), Tree('c', [Tree('d')])])\nprint('Tree structure:')
\nprint(f'Root: {tree.label}')\nprint(f'Children: {[child.label for
child in tree.children]}')\nprint(f'Child 0 children: {[child.label for
child in tree.children[0].children]}')\nprint(f'Child 1 children:
{[child.label for child in tree.children[1].children]}')\nif
tree.children[1].children:\n    print(f'Grandchild children:
{[child.label for child in tree.children[1].children[0].children]}')
\n\ntry:\n    path = tree.path_to('b', 'd')\n    print('\\nPath found:',
path)\nexcept ValueError as e:\n    print('\\nError raised:', str(e))
\n\n# Let's also check if both nodes exist\nfrom main import Tree\n\ndef
find_node(node, target):\n    if node.label == target:\n        return
node\n    for child in node.children:\n        result = find_node(child,
target)\n        if result:\n            return result\n    return
None\n\nb_exists = find_node(tree, 'b') is not None\nd_exists =
find_node(tree, 'd') is not None\nprint(f'\\nNode b exists: {b_exists}')
\nprint(f'Node d exists: {d_exists}')", 'file_path': './debug_path.py'}
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:40,213 - agent -
INFO - Tool observation (first 300 chars): Tree structure:
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Root: a
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Children: ['b', 'c']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 0 children: []
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Child 1 children: ['d']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Grandchild children: []
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Path found: ['b', 'a', 'c', 'd']
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Node b exists: True
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] Node d exists: True
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:40,214 - agent -
WARNING - Workflow ended by reaching max steps (30).
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:40,214 - agent -
INFO - Workflow execution complete. Generating final patch.
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:40,214 - agent -
INFO - Generating final git patch...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:40,236 - agent -
INFO - Generating final git patch...
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] ✅ CREATE WORKFLOW COMPLETED
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:40,256 - agent -
INFO - [WORKFLOW] CREATE workflow completed
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] HEAD is now at af216df Initial
commit
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 🏁 AGENT FINISHED
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] 2025-10-21 19:34:40,261 - agent -
INFO - [AGENT-END] Agent execution completed
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Exited agent's
agent_main()
```

```
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Writing output.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Wrote output.json
DEBUG: [DOCKER:sandbox_tmpf8qgj5x7] [AGENT_RUNNER] Exiting main()
DEBUG: [SANDBOX] <sandbox_tmpf8qgj5x7> finished running
DEBUG: [SANDBOX] <sandbox_tmpf8qgj5x7> captured 484 lines of logs
DEBUG: [SANDBOX] Read output.json from <sandbox_tmpf8qgj5x7>: /tmp/tmpf8qgj5x7/output.json
DEBUG: [PROBLEM_SUITE] Validating diff generated by <sandbox_tmpf8qgj5x7> for pov
DEBUG: [PROBLEM_SUITE] Diff generated by <sandbox_tmpf8qgj5x7> for pov is valid
 INFO: [PROBLEM_SUITE] Finished sandbox to run agent for problem pov: success
```

```
========== DIFF (322 lines) ==========
diff --git a/debug_path.py b/debug_path.py
new file mode 100644
index 0000000..b868d33
--- /dev/null
+++ b/debug_path.py
@@ -0,0 +1,34 @@
+from main import Tree
+
+# Debug the path finding issue
+tree = Tree('a', [Tree('b'), Tree('c', [Tree('d')])])
+print('Tree structure:')
+print(f'Root: {tree.label}')
+print(f'Children: {[child.label for child in tree.children]}')
+print(f'Child 0 children: {[child.label for child in tree.children[0].children]}')
+print(f'Child 1 children: {[child.label for child in tree.children[1].children]}')
+if tree.children[1].children:
+    print(f'Grandchild children: {[child.label for child in tree.children[1].children[0].children]}')
+
+try:
+    path = tree.path_to('b', 'd')
+    print('\nPath found:', path)
+except ValueError as e:
+    print('\nError raised:', str(e))
+
+# Let's also check if both nodes exist
+from main import Tree
+
+def find_node(node, target):
+    if node.label == target:
+        return node
+    for child in node.children:
+        result = find_node(child, target)
+        if result:
+            return result
+    return None
+
+b_exists = find_node(tree, 'b') is not None
+d_exists = find_node(tree, 'd') is not None
+print(f'\nNode b exists: {b_exists}')
+print(f'Node d exists: {d_exists}')
\ No newline at end of file
diff --git a/debug_test.py b/debug_test.py
new file mode 100644
index 0000000..1e93d69
--- /dev/null
+++ b/debug_test.py
@@ -0,0 +1,45 @@
+from main import Tree
+
+# Test case 5: from_pov('d')
+tree = Tree('a', [Tree('b', [Tree('c'), Tree('d')]), Tree('e',
[Tree('f'), Tree('g')])])
+expected = Tree('d', [Tree('b', [Tree('a', [Tree('e', [Tree('f'),
Tree('g')]), Tree('c')])])])
+result = tree.from_pov('d')
+
+print('=== Test Case 5 ===')
+print('Expected structure:')
+print(f'Root: {expected.label}')
+print(f'Children: {[child.label for child in expected.children]}')
+if expected.children:
+    print(f'Child 0 children: {[child.label for child in
expected.children[0].children]}')
+    if expected.children[0].children:
+        print(f'Grandchild 0 children: {[child.label for child in
expected.children[0].children[0].children]}')
+
+print('\nActual structure:')
+print(f'Root: {result.label}')
+print(f'Children: {[child.label for child in result.children]}')
+if result.children:
+    print(f'Child 0 children: {[child.label for child in
result.children[0].children]}')
+    if result.children[0].children:
+        print(f'Grandchild 0 children: {[child.label for child in
result.children[0].children[0].children]}')
+
+print('\nEqual:', expected == result)
+
+# Test case 6: from_pov('e')
+tree2 = Tree('a', [Tree('b', [Tree('c'), Tree('d')]), Tree('e',
[Tree('f'), Tree('g')])])
+expected2 = Tree('e', [Tree('a', [Tree('b', [Tree('c'), Tree('d')]),
Tree('f'), Tree('g')])])
+result2 = tree2.from_pov('e')
+
+print('\n=== Test Case 6 ===')
+print('Expected structure:')
+print(f'Root: {expected2.label}')
+print(f'Children: {[child.label for child in expected2.children]}')
+if expected2.children:
```

```
+    print(f'Child 0 children: {[child.label for child in
expected2.children[0].children]}')
+
+print('\nActual structure:')
+print(f'Root: {result2.label}')
+print(f'Children: {[child.label for child in result2.children]}')
+if result2.children:
+    print(f'Child 0 children: {[child.label for child in
result2.children[0].children]}')
+
+print('\nEqual:', expected2 == result2)
\ No newline at end of file
diff --git a/main.py b/main.py
index 21dbe4b..c6c45b2 100644
--- a/main.py
+++ b/main.py
@@ -19,7 +19,107 @@ class Tree:
         return self.__dict__() == other.__dict__()
 
     def from_pov(self, from_node):
-        pass
+        # First, let's find if the tree contains the from_node
+        def find_node(node, target):
+            if node.label == target:
+                return node
+            for child in node.children:
+                result = find_node(child, target)
+                if result:
+                    return result
+            return None
+
+        # Build parent map to navigate upwards
+        def build_parent_map(node, parent=None):
+            parent_map[node.label] = parent
+            for child in node.children:
+                build_parent_map(child, node)
+
+        # Check if from_node exists in the tree
+        target_node = find_node(self, from_node)
+        if not target_node:
+            raise ValueError("Tree could not be reoriented")
+
+        # Build parent map for the entire tree
+        parent_map = {}
+        build_parent_map(self)
+
+        # Build a map of all nodes and their original children
+        def build_children_map(node):
+            children_map[node.label] = [child.label for child in
node.children]
+            for child in node.children:
+                build_children_map(child)
+
+        children_map = {}
+        build_children_map(self)
+
+        # Build the new tree using a simpler approach
+        # Start from the new root and recursively build the tree
+        visited = set()
+
+        def build_tree_recursive(current_label, came_from=None):
+            if current_label in visited:
+                return None
+            visited.add(current_label)
+
+            new_children = []
+
+            # Add original children (except the one we came from)
+            for child_label in children_map.get(current_label, []):
+                if child_label != came_from:
+                    child_tree = build_tree_recursive(child_label,
current_label)
+                    if child_tree:
+                        new_children.append(child_tree)
+
+            # Add parent as a child (except the one we came from)
+            parent = parent_map.get(current_label)
+            if parent and parent.label != came_from:
+                parent_tree = build_tree_recursive(parent.label,
current_label)
+                if parent_tree:
+                    new_children.append(parent_tree)
+
+            return Tree(current_label, new_children)
+
+        return build_tree_recursive(from_node)

     def path_to(self, from_node, to_node):
-        pass
+        # First check if both nodes exist in the tree
+        def find_node(node, target):
+            if node.label == target:
+                return node
+            for child in node.children:
+                result = find_node(child, target)
+                if result:
+                    return result
+            return None
+
+        from_exists = find_node(self, from_node) is not None
+        to_exists = find_node(self, to_node) is not None
+
+        if not from_exists or not to_exists:
+            raise ValueError("No path found")
+
+        # Get tree from from_node's perspective
+        try:
+            new_tree = self.from_pov(from_node)
+        except ValueError:
+            raise ValueError("No path found")
+
+        # Now find path in the new tree
+        def find_path(node, target, path):
+            path.append(node.label)
+            if node.label == target:
+                return path[:]
+
+            for child in node.children:
+                result = find_path(child, target, path)
+                if result:
+                    return result
+
+            path.pop()
+            return None
+
+        path = find_path(new_tree, to_node, [])
+        if path is None:
+            raise ValueError("No path found")
+        return path
\ No newline at end of file
diff --git a/test_tree_test.py b/test_tree_test.py
new file mode 100644
index 0000000..3b66e0d
--- /dev/null
+++ b/test_tree_test.py
@@ -0,0 +1,107 @@
+import unittest
+
+from main import Tree
+
+
+class TreeTest(unittest.TestCase):
+    def test_empty_tree(self):
+        tree = Tree('a')
+        expected = Tree('a')
+        self.assertEqual(tree.from_pov('a'), expected)
+
+    def test_single_node_tree(self):
+        tree = Tree('a')
+        expected = Tree('a')
+        self.assertEqual(tree.from_pov('a'), expected)
+
+    def test_multi_node_tree(self):
+        tree = Tree('a', [Tree('b'), Tree('c')])
+        expected = Tree('b', [Tree('a', [Tree('c')])])
+        self.assertEqual(tree.from_pov('b'), expected)
+
+    def test_multi_node_tree_2(self):
+        tree = Tree('a', [Tree('b', [Tree('c')]), Tree('d')])
+        expected = Tree('c', [Tree('b', [Tree('a', [Tree('d')])])])
+        self.assertEqual(tree.from_pov('c'), expected)
+
+    def test_multi_node_tree_3(self):
+        tree = Tree('a', [Tree('b', [Tree('c', [Tree('d')])])])
+        expected = Tree('d', [Tree('c', [Tree('b', [Tree('a')])])])
+        self.assertEqual(tree.from_pov('d'), expected)
+
+    def test_multi_node_tree_4(self):
```

```
+        tree = Tree('a', [Tree('b'), Tree('c'), Tree('d')])
+        expected = Tree('b', [Tree('a', [Tree('c'), Tree('d')])])
+        self.assertEqual(tree.from_pov('b'), expected)
+
+    def test_multi_node_tree_5(self):
+        tree = Tree('a', [Tree('b', [Tree('c'), Tree('d')]), Tree('e',
[Tree('f'), Tree('g')])])
+        expected = Tree('d', [Tree('b', [Tree('a', [Tree('e',
[Tree('f'), Tree('g')]), Tree('c')])])])
+        self.assertEqual(tree.from_pov('d'), expected)
+
+    def test_multi_node_tree_6(self):
+        tree = Tree('a', [Tree('b', [Tree('c'), Tree('d')]), Tree('e',
[Tree('f'), Tree('g')])])
+        expected = Tree('e', [Tree('a', [Tree('b', [Tree('c'),
Tree('d')]), Tree('f'), Tree('g')])])
+        self.assertEqual(tree.from_pov('e'), expected)
+
+    def test_multi_node_tree_7(self):
+        tree = Tree('a', [Tree('b', [Tree('c'), Tree('d')]), Tree('e',
[Tree('f'), Tree('g')])])
+        expected = Tree('a', [Tree('b', [Tree('c'), Tree('d')]),
Tree('e', [Tree('f'), Tree('g')])])
+        self.assertEqual(tree.from_pov('a'), expected)
+
+    def test_node_not_found(self):
+        tree = Tree('a', [Tree('b'), Tree('c')])
+        with self.assertRaises(ValueError) as context:
+            tree.from_pov('d')
+        self.assertEqual(str(context.exception), "Tree could not be
reoriented")
+
+    def test_path_to_self(self):
+        tree = Tree('a')
+        expected = ['a']
+        self.assertEqual(tree.path_to('a', 'a'), expected)
+
+    def test_path_to_sibling(self):
+        tree = Tree('a', [Tree('b'), Tree('c')])
+        expected = ['b', 'a', 'c']
+        self.assertEqual(tree.path_to('b', 'c'), expected)
+
+    def test_path_to_parent(self):
+        tree = Tree('a', [Tree('b'), Tree('c')])
+        expected = ['b', 'a']
+        self.assertEqual(tree.path_to('b', 'a'), expected)
+
+    def test_path_to_child(self):
+        tree = Tree('a', [Tree('b'), Tree('c')])
+        expected = ['a', 'b']
+        self.assertEqual(tree.path_to('a', 'b'), expected)
+
+    def test_path_to_grandchild(self):
+        tree = Tree('a', [Tree('b', [Tree('c')]), Tree('d')])
+        expected = ['a', 'b', 'c']
+        self.assertEqual(tree.path_to('a', 'c'), expected)
```

```
+
+    def test_path_to_cousin(self):
+        tree = Tree('a', [Tree('b', [Tree('c'), Tree('d')]), Tree('e',
[Tree('f'), Tree('g')])])
+        expected = ['c', 'b', 'a', 'e', 'f']
+        self.assertEqual(tree.path_to('c', 'f'), expected)
+
+    def test_path_to_nonexistent_node(self):
+        tree = Tree('a', [Tree('b'), Tree('c')])
+        with self.assertRaises(ValueError) as context:
+            tree.path_to('a', 'd')
+        self.assertEqual(str(context.exception), "No path found")
+
+    def test_path_from_nonexistent_node(self):
+        tree = Tree('a', [Tree('b'), Tree('c')])
+        with self.assertRaises(ValueError) as context:
+            tree.path_to('d', 'a')
+        self.assertEqual(str(context.exception), "No path found")
+
+    def test_path_between_unconnected_nodes(self):
+        tree = Tree('a', [Tree('b'), Tree('c', [Tree('d')])])
+        with self.assertRaises(ValueError) as context:
+            tree.path_to('b', 'd')
+        self.assertEqual(str(context.exception), "No path found")
+
+if __name__ == '__main__':
+    unittest.main()
\ No newline at end of file

========== LOGS (484 lines) ==========
```

```
 INFO: [PROBLEM_SUITE] Starting sandbox to evaluate solution diff for
problem pov
DEBUG: [SANDBOX] Created sandbox temp directory for
<sandbox_tmpg_v64m7s>: /tmp/tmpg_v64m7s
DEBUG: [POLYGLOT] Copied main.py to /tmp/tmpg_v64m7s/repo for pov
DEBUG: [POLYGLOT] Copied tests.py to /tmp/tmpg_v64m7s/repo for pov
DEBUG: [POLYGLOT] Initializing git repository in /tmp/tmpg_v64m7s/repo
for pov
DEBUG: [GIT] Initializing git repository in /tmp/tmpg_v64m7s/repo
DEBUG: [GIT] Initialized git repository in /tmp/tmpg_v64m7s/repo
DEBUG: [GIT] Adding all files in /tmp/tmpg_v64m7s/repo
DEBUG: [GIT] Added all files in /tmp/tmpg_v64m7s/repo
DEBUG: [GIT] Making initial commit: Initial commit
DEBUG: [GIT] Made initial commit: Initial commit
DEBUG: [POLYGLOT] Initialized git repository in /tmp/tmpg_v64m7s/repo
for pov
DEBUG: [PROBLEM_SUITE] Applying agent's solution diff to /tmp/
tmpg_v64m7s/repo for problem pov
DEBUG: [PROBLEM_SUITE] Applied agent's solution diff to /tmp/
tmpg_v64m7s/repo for problem pov
DEBUG: [SANDBOX] Copied main Python script (/root/abstract-agent-runner/
problem_suites/polyglot/TEST_RUNNER.py) for <sandbox_tmpg_v64m7s>: /tmp/
tmpg_v64m7s/TEST_RUNNER.py
```

```
DEBUG: [SANDBOX] Written input.json for <sandbox_tmpg_v64m7s>: /tmp/
tmpg_v64m7s/input.json
DEBUG: [SANDBOX] Running sandbox <sandbox_tmpg_v64m7s>
DEBUG: [SANDBOX] Started sandbox runner thread for <sandbox_tmpg_v64m7s>
DEBUG: [PROBLEM_SUITE] Started sandbox to evaluate solution diff for
problem pov
DEBUG: [SANDBOX] Cleaned up sandbox <sandbox_tmpf8qgj5x7>
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Entered
main()
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Loading
main.py
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Loaded
main.py
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Loading
tests.py
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Loaded
tests.py
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Found test
class: PovTest
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Found 15 test
methods
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [1/15]
Running test_can_find_path_from_nodes_other_than_x...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_find_path_from_nodes_other_than_x: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [2/15]
Running test_can_find_path_not_involving_root...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_find_path_not_involving_root: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [3/15]
Running test_can_find_path_to_cousin...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_find_path_to_cousin: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [4/15]
Running test_can_find_path_to_parent...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_find_path_to_parent: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [5/15]
Running test_can_find_path_to_sibling...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_find_path_to_sibling: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [6/15]
Running test_can_reroot_a_complex_tree_with_cousins...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_reroot_a_complex_tree_with_cousins: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [7/15]
Running test_can_reroot_a_tree_with_a_parent_and_many_siblings...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_reroot_a_tree_with_a_parent_and_many_siblings: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [8/15]
Running test_can_reroot_a_tree_with_a_parent_and_one_sibling...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_reroot_a_tree_with_a_parent_and_one_sibling: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [9/15]
Running test_can_reroot_a_tree_with_new_root_deeply_nested_in_tree...
```

```
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_can_reroot_a_tree_with_new_root_deeply_nested_in_tree: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [10/15]
Running test_errors_if_destination_does_not_exist...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_errors_if_destination_does_not_exist: PASSED
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] [11/15]
Running test_errors_if_source_does_not_exist...
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER]
test_errors_if_source_does_not_exist: FAILED - 'No path found' != 'Tree
could not be reoriented'
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] - No path found
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] + Tree could not be reoriented
DEBUG: [DOCKER:sandbox_tmpg_v64m7s]
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Test results:
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [{'name':
'test_can_find_path_from_nodes_other_than_x', 'status': 'pass'},
{'name': 'test_can_find_path_not_involving_root', 'status': 'pass'},
{'name': 'test_can_find_path_to_cousin', 'status': 'pass'}, {'name':
'test_can_find_path_to_parent', 'status': 'pass'}, {'name':
'test_can_find_path_to_sibling', 'status': 'pass'}, {'name':
'test_can_reroot_a_complex_tree_with_cousins', 'status': 'pass'},
{'name': 'test_can_reroot_a_tree_with_a_parent_and_many_siblings',
'status': 'pass'}, {'name':
'test_can_reroot_a_tree_with_a_parent_and_one_sibling', 'status':
'pass'}, {'name':
'test_can_reroot_a_tree_with_new_root_deeply_nested_in_tree', 'status':
'pass'}, {'name': 'test_errors_if_destination_does_not_exist', 'status':
'pass'}, {'name': 'test_errors_if_source_does_not_exist', 'status':
'fail'}, {'name':
'test_errors_if_target_does_not_exist_in_a_large_tree', 'status':
'skip'}, {'name':
'test_errors_if_target_does_not_exist_in_a_singleton_tree', 'status':
'skip'}, {'name':
'test_moves_children_of_the_new_root_to_same_level_as_former_parent',
'status': 'skip'}, {'name':
'test_results_in_the_same_tree_if_the_input_tree_is_a_singleton',
'status': 'skip'}]
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Test summary:
10 passed, 1 failed, 4 skipped
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Writing
output.json
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Wrote
output.json
DEBUG: [DOCKER:sandbox_tmpg_v64m7s] [POLYGLOT_TEST_RUNNER] Exiting
main()
DEBUG: [SANDBOX] <sandbox_tmpg_v64m7s> finished running
DEBUG: [SANDBOX] <sandbox_tmpg_v64m7s> captured 38 lines of logs
DEBUG: [SANDBOX] Read output.json from <sandbox_tmpg_v64m7s>: /tmp/
tmpg_v64m7s/output.json
 INFO: [PROBLEM_SUITE] Finished sandbox to evaluate solution diff for
problem pov: success
```

```
========== TEST RESULTS ==========
10 passed, 1 failed, 4 skipped
test_can_find_path_from_nodes_other_than_x - no category - pass
test_can_find_path_not_involving_root - no category - pass
test_can_find_path_to_cousin - no category - pass
test_can_find_path_to_parent - no category - pass
test_can_find_path_to_sibling - no category - pass
test_can_reroot_a_complex_tree_with_cousins - no category - pass
test_can_reroot_a_tree_with_a_parent_and_many_siblings - no category -
pass
test_can_reroot_a_tree_with_a_parent_and_one_sibling - no category -
pass
test_can_reroot_a_tree_with_new_root_deeply_nested_in_tree - no category
- pass
test_errors_if_destination_does_not_exist - no category - pass
test_errors_if_source_does_not_exist - no category - fail
test_errors_if_target_does_not_exist_in_a_large_tree - no category -
skip
test_errors_if_target_does_not_exist_in_a_singleton_tree - no category -
skip
test_moves_children_of_the_new_root_to_same_level_as_former_parent - no
category - skip
test_results_in_the_same_tree_if_the_input_tree_is_a_singleton - no
category - skip
========== LOGS (38 lines) ==========



DEBUG: [SANDBOX] Cleaned up sandbox <sandbox_tmpg_v64m7s>
```

```
(.venv_clean) root@kind-name-wilts-fin-01:~/abstract-agent-runner#
python cli.py polyglot react  agent-maxi-2.py http://135.181.71.13:8000
--log-docker-to-stdout --verbose --timeout 1200
 INFO: [POLYGLOT] Loaded 33 problems from datasets/polyglot/
polyglot.json
DEBUG: [POLYGLOT]     Problem affine-cipher verified successfully (found
16 associated tests)
DEBUG: [POLYGLOT]     Problem beer-song verified successfully (found 8
associated tests)
DEBUG: [POLYGLOT]     Problem book-store verified successfully (found 20
associated tests)
DEBUG: [POLYGLOT]     Problem bottle-song verified successfully (found 7
associated tests)
DEBUG: [POLYGLOT]     Problem bowling verified successfully (found 31
associated tests)
DEBUG: [POLYGLOT]     Problem connect verified successfully (found 10
associated tests)
DEBUG: [POLYGLOT]     Problem dominoes verified successfully (found 13
associated tests)
```

```
DEBUG: [POLYGLOT]     Problem dot-dsl verified successfully (found 12
associated tests)
DEBUG: [POLYGLOT]     Problem food-chain verified successfully (found 10
associated tests)
DEBUG: [POLYGLOT]     Problem forth verified successfully (found 54
associated tests)
DEBUG: [POLYGLOT]     Problem go-counting verified successfully (found
11 associated tests)
DEBUG: [POLYGLOT]     Problem grade-school verified successfully (found
20 associated tests)
DEBUG: [POLYGLOT]     Problem grep verified successfully (found 25
associated tests)
DEBUG: [POLYGLOT]     Problem hangman verified successfully (found 7
associated tests)
DEBUG: [POLYGLOT]     Problem list-ops verified successfully (found 24
associated tests)
DEBUG: [POLYGLOT]     Problem phone-number verified successfully (found
21 associated tests)
DEBUG: [POLYGLOT]     Problem pig-latin verified successfully (found 22
associated tests)
DEBUG: [POLYGLOT]     Problem poker verified successfully (found 37
associated tests)
DEBUG: [POLYGLOT]     Problem pov verified successfully (found 15
associated tests)
DEBUG: [POLYGLOT]     Problem proverb verified successfully (found 8
associated tests)
DEBUG: [POLYGLOT]     Problem react verified successfully (found 14
associated tests)
DEBUG: [POLYGLOT]     Problem rest-api verified successfully (found 9
associated tests)
DEBUG: [POLYGLOT]     Problem robot-name verified successfully (found 4
associated tests)
DEBUG: [POLYGLOT]     Problem scale-generator verified successfully
(found 17 associated tests)
DEBUG: [POLYGLOT]     Problem sgf-parsing verified successfully (found
23 associated tests)
DEBUG: [POLYGLOT]     Problem simple-linked-list verified successfully
(found 20 associated tests)
DEBUG: [POLYGLOT]     Problem transpose verified successfully (found 12
associated tests)
DEBUG: [POLYGLOT]     Problem tree-building verified successfully (found
13 associated tests)
DEBUG: [POLYGLOT]     Problem two-bucket verified successfully (found 9
associated tests)
DEBUG: [POLYGLOT]     Problem variable-length-quantity verified
successfully (found 26 associated tests)
DEBUG: [POLYGLOT]     Problem wordy verified successfully (found 25
associated tests)
DEBUG: [POLYGLOT]     Problem zebra-puzzle verified successfully (found
2 associated tests)
DEBUG: [POLYGLOT]     Problem zipper verified successfully (found 14
associated tests)
 INFO: [POLYGLOT] Successfully loaded 33 problems
 INFO: Problem react has 14 tests
 INFO: [SANDBOX] Checking gateway URL: http://135.181.71.13:8000
 INFO: [SANDBOX] Gateway URL http://135.181.71.13:8000 is valid
```

```
DEBUG: [SANDBOX] Stopping and deleting all containers
DEBUG: [SANDBOX] Stopped and deleted all containers
 INFO: [SANDBOX] Building Docker image: sandbox-image
[+] Building 1.0s (11/11) FINISHED
docker:default
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 1.04kB
0.0s
 => [internal] load metadata for docker.io/library/python:3.11-slim
0.9s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [1/6] FROM docker.io/library/python:3.11-
slim@sha256:b6000fc45f769f42c4c717dab2675bbb0ec6531c32a0483a2f78de0b7023
e71b
0.0s
 => => resolve docker.io/library/python:3.11-
slim@sha256:b6000fc45f769f42c4c717dab2675bbb0ec6531c32a0483a2f78de0b7023
e71b
0.0s
 => [internal] load build context
0.0s
 => => transferring context: 46B
0.0s
 => CACHED [2/6] RUN apt-get update &&     apt-get install -y --no-
install-recommends git patch diffutils &&     rm -rf /var/lib/apt/lists/
*
0.0s
 => CACHED [3/6] COPY sandbox_requirements.txt /tmp/
sandbox_requirements.txt
0.0s
 => CACHED [4/6] RUN pip install --no-cache-dir --upgrade pip &&     pip
install --no-cache-dir -r /tmp/sandbox_requirements.txt &&     rm /tmp/
sandbox_requirements.txt                                         0.0s
 => CACHED [5/6] RUN pip cache purge &&     rm -rf /root/.cache/pip
0.0s
 => CACHED [6/6] WORKDIR /sandbox
0.0s
 => exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:6a0d37b7a80c209efc720b51333a5af41822e44e249d5749ba8c3dcb2762dbcd
0.0s
 => => naming to docker.io/library/sandbox-image
0.0s
 INFO: [SANDBOX] Successfully built Docker image: sandbox-image
DEBUG: [SANDBOX] Found sandbox network: sandbox-network
 INFO: [SANDBOX] Building Docker image: sandbox-proxy-image
[+] Building 1.0s (8/8) FINISHED
docker:default
```

```
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 289B
0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine
0.9s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [1/3] FROM docker.io/library/
nginx:alpine@sha256:61e01287e546aac28a3f56839c136b31f590273f3b41187a36f4
6f6a03bbfe22
0.0s
 => [internal] load build context
0.0s
 => => transferring context: 41B
0.0s
 => CACHED [2/3] RUN apk add --no-cache gettext
0.0s
 => CACHED [3/3] COPY nginx.conf.template /tmp/nginx.conf.template
0.0s
 => exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:b38edefd62e5475e622d31caf7dabe385f28d175b050c8ae714b02c9758742a3
0.0s
 => => naming to docker.io/library/sandbox-proxy-image
0.0s
 INFO: [SANDBOX] Successfully built Docker image: sandbox-proxy-image
 INFO: [SANDBOX] Running sandbox proxy
DEBUG: [SANDBOX] Connected sandbox proxy to bridge network
DEBUG: [SANDBOX] Starting watchdog thread
DEBUG: [SANDBOX] Started watchdog thread
 INFO: [PROBLEM_SUITE] Starting sandbox to run agent for problem react
DEBUG: [SANDBOX] Created sandbox temp directory for
<sandbox_tmp862lkupv>: /tmp/tmp862lkupv
DEBUG: [POLYGLOT] Copied main.py to /tmp/tmp862lkupv/repo for react
DEBUG: [POLYGLOT] Initializing git repository in /tmp/tmp862lkupv/repo
for react
DEBUG: [GIT] Initializing git repository in /tmp/tmp862lkupv/repo
DEBUG: [GIT] Initialized git repository in /tmp/tmp862lkupv/repo
DEBUG: [GIT] Adding all files in /tmp/tmp862lkupv/repo
DEBUG: [GIT] Added all files in /tmp/tmp862lkupv/repo
DEBUG: [GIT] Making initial commit: Initial commit
DEBUG: [GIT] Made initial commit: Initial commit
DEBUG: [POLYGLOT] Initialized git repository in /tmp/tmp862lkupv/repo
for react
DEBUG: [SANDBOX] Copied main Python script (/root/abstract-agent-runner/
problem_suites/AGENT_RUNNER.py) for <sandbox_tmp862lkupv>: /tmp/
tmp862lkupv/AGENT_RUNNER.py
DEBUG: [SANDBOX] Written input.json for <sandbox_tmp862lkupv>: /tmp/
tmp862lkupv/input.json
DEBUG: [SANDBOX] Running sandbox <sandbox_tmp862lkupv>
```

```
DEBUG: [SANDBOX] Started sandbox runner thread for <sandbox_tmp862lkupv>
DEBUG: [PROBLEM_SUITE] Started sandbox to run agent for problem react
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Entered main()
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Reading input.json
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Read input.json
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Loading /sandbox/
agent.py
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Loaded /sandbox/
agent.py
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] agent_main() function
found in /sandbox/agent.py
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Entering agent's
agent_main()
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmp862lkupv] 🚀 AGENT STARTING
DEBUG: [DOCKER:sandbox_tmp862lkupv]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:58,250 - agent -
INFO - [AGENT-START] Run ID: 8366b657-e085-4a65-b899-ef494c437792
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:58,250 - agent -
INFO - [AGENT-START] Repo directory: /sandbox/repo
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:58,250 - agent -
INFO - [AGENT-START] Test mode: False
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:58,250 - agent -
INFO - [AGENT-START] Timeout: 1800s, Max steps: 400
DEBUG: [DOCKER:sandbox_tmp862lkupv] [DEBUG] Starting git initialization
check...
DEBUG: [DOCKER:sandbox_tmp862lkupv] [DEBUG] Work directory: /sandbox/
repo
DEBUG: [DOCKER:sandbox_tmp862lkupv] [DEBUG] Before chdir - pwd shows: /
sandbox/repo
DEBUG: [DOCKER:sandbox_tmp862lkupv] [DEBUG] After chdir - pwd shows: /
sandbox/repo
DEBUG: [DOCKER:sandbox_tmp862lkupv] [DEBUG] Git repository already
exists
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:58,258 - agent -
INFO - [AGENT-START] Problem statement length: 728 chars
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmp862lkupv] 🔍 DETERMINING PROBLEM TYPE...
DEBUG: [DOCKER:sandbox_tmp862lkupv]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:58,258 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 171/219340
tokens
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,675 - agent -
INFO - [PROBLEM-TYPE] Determined: CREATE
DEBUG: [DOCKER:sandbox_tmp862lkupv] ✅ Problem type: CREATE
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmp862lkupv] 🆕 CREATE TASK DETECTED - STARTING
CREATE WORKFLOW
DEBUG: [DOCKER:sandbox_tmp862lkupv]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,675 - agent -
INFO - [WORKFLOW] Starting CREATE task workflow
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]
┌───────────────────────────────────────────────────────────────────────
───────┐
DEBUG: [DOCKER:sandbox_tmp862lkupv] │ 🆕 CREATE TASK WORKFLOW STARTING
│
DEBUG: [DOCKER:sandbox_tmp862lkupv]
└───────────────────────────────────────────────────────────────────────
───────┘
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,675 - agent -
INFO - [CREATE] Starting CREATE task workflow
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,675 - agent -
INFO - [CREATE] Problem statement length: 728 chars
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,675 - agent -
INFO - [CREATE] Step 1: Generating code skeleton
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,676 - agent -
INFO - [CREATE] Code skeleton generated: 304 chars
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,676 - agent -
INFO - [CREATE] Step 2: Generating initial solution
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,676 - agent -
INFO - Starting multi-step reasoning solution generation
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:38:59,676 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 618/219340
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:02,285 - agent -
INFO - Multi-step reasoning solution generation completed successfully
with infinite loop validation
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:02,285 - agent -
INFO - Generated initial solution successfully using multi-step
reasoning
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:02,285 - agent -
INFO - [CREATE] Initial solution generated: 1743 chars
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:02,285 - agent -
INFO - [CREATE] Step 3: Extracting and writing solution files
DEBUG: [DOCKER:sandbox_tmp862lkupv] Created file: ./main.py
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:02,285 - agent -
INFO - [CREATE] Created 1 solution files
DEBUG: [DOCKER:sandbox_tmp862lkupv] ✅ Created 1 solution files
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:02,285 - agent -
INFO - [CREATE] Step 4: Generating test files
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:02,285 - agent -
INFO - Starting test cases generation
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:02,285 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 332/219340
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:15,579 - agent -
INFO - Step 1 - Testcase Generation completed
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:15,579 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 652/219340
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,983 - agent -
INFO - Step 2 - Testcase check completed
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,983 - agent -
INFO - Multi-step reasoning solution generation completed successfully
with infinite loop validation
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,983 - agent -
INFO - Generated testcases successfully using multi-step reasoning
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,983 - agent -
INFO - [CREATE] Test cases generated: 5412 chars
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,983 - agent -
INFO - [CREATE] Step 5: Extracting and writing test files
DEBUG: [DOCKER:sandbox_tmp862lkupv] Created file: ./
test_reactive_system.py
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,983 - agent -
INFO - [CREATE] Created 1 test files
DEBUG: [DOCKER:sandbox_tmp862lkupv] ✅ Created 1 test files
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,983 - agent -
INFO - [CREATE] Time spent: 25.3s, Remaining timeout: 1714.7s
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,983 - agent -
INFO - [CREATE] Step 6: Starting iterative FIX workflow for refinement
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmp862lkupv] 🔧 Starting iterative refinement
workflow...
DEBUG: [DOCKER:sandbox_tmp862lkupv]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]
┌───────────────────────────────────────────────────────────────────────
───────┐
DEBUG: [DOCKER:sandbox_tmp862lkupv] │ 🔧 ITERATIVE FIX WORKFLOW STARTING
│
DEBUG: [DOCKER:sandbox_tmp862lkupv]
└───────────────────────────────────────────────────────────────────────
───────┘
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,984 - agent -
INFO - [ITERATIVE] Starting iterative FIX workflow
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,984 - agent -
INFO - [ITERATIVE] Run ID: 8366b657-e085-4a65-b899-ef494c437792
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,984 - agent -
INFO - [ITERATIVE] Max steps: 30, Timeout: 1714.691801071167s
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,984 - agent -
INFO - [ITERATIVE] Test runner: unittest (mode: FILE)
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,984 - agent -
INFO - State file not found at .agent_state.json. Starting with a fresh
state.
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,984 - agent -
INFO - [ITERATIVE] Loaded COT state: 0 existing thoughts
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,986 - agent -
INFO - [ITERATIVE] Initialized tool manager with 13 tools
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,986 - agent -
INFO - [ITERATIVE] Workflow started at 19:39:24
DEBUG: [DOCKER:sandbox_tmp862lkupv] 🚀 Starting iterative workflow (max
30 steps, 1714.691801071167s timeout)
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,986 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 1/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2364/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:24,986 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2364/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:31,218 - agent -
INFO - LLM proposed action: search_in_all_files_content with args:
{'search_term': 'reactive'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:31,219 - agent -
INFO - Tool observation (first 300 chars): ./
test_reactive_system.py:4:class
TestReactiveSystem(unittest.TestCase):...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:31,219 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 2/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2429/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:31,220 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2429/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:35,494 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./test_reactive_system.py'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:35,494 - agent -
INFO - Tool observation (first 300 chars): import unittest
DEBUG: [DOCKER:sandbox_tmp862lkupv] from main import InputCell,
ComputeCell
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv] class
TestReactiveSystem(unittest.TestCase):
DEBUG: [DOCKER:sandbox_tmp862lkupv]
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv]     def
test_input_cells_have_a_value(self):
DEBUG: [DOCKER:sandbox_tmp862lkupv]         input_cell = InputCell(10)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
self.assertEqual(input_cell.value, 10)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]     def
test_compute_cells_calculate_initial_value(self):
DEBUG: [DOCKER:sandbox_tmp862lkupv]    ...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:35,494 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 3/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2853/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:35,495 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2853/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:38,485 - agent -
INFO - LLM proposed action: search_in_all_files_content with args:
{'search_term': 'main.py'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:38,486 - agent -
INFO - Tool observation (first 300 chars): 'main.py' not found in the
codebase....
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:38,486 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 4/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2918/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:38,486 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2918/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:41,718 - agent -
INFO - LLM proposed action: search_in_all_files_content with args:
{'search_term': '.py'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:41,719 - agent -
INFO - Tool observation (first 300 chars): '.py' not found in the
codebase....
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:41,719 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 5/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2986/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:41,719 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 2986/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:44,391 - agent -
INFO - LLM proposed action: search_in_all_files_content with args:
{'search_term': 'class'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:44,392 - agent -
INFO - Tool observation (first 300 chars): ./main.py:1:class InputCell:
DEBUG: [DOCKER:sandbox_tmp862lkupv] ./main.py:15:class ComputeCell:
DEBUG: [DOCKER:sandbox_tmp862lkupv] ./test_reactive_system.py:4:class
TestReactiveSystem(unittest.TestCase):...
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:44,392 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 6/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3035/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:44,393 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3035/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:47,413 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./main.py'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:47,413 - agent -
INFO - Tool observation (first 300 chars): class InputCell:
DEBUG: [DOCKER:sandbox_tmp862lkupv]     def __init__(self,
initial_value):
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.value = initial_value
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.dependents = set()
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.callbacks = set()
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]     def set_value(self, new_value):
DEBUG: [DOCKER:sandbox_tmp862lkupv]         if self.value != new_value:
DEBUG: [DOCKER:sandbox_tmp862lkupv]             self.value = new_value
DEBUG: [DOCKER:sandbox_tmp862lkupv]             # Notify all dependents
of...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:47,414 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:47,414 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 7/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3228/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:47,414 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3228/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:50,312 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_reactive_system.py']}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:50,312 - agent -
INFO - Running test command: python -m unittest test_reactive_system
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:50,428 - agent -
INFO - Tool observation (first 300 chars): EEE.EEEEEE.E
DEBUG: [DOCKER:sandbox_tmp862lkupv]
======================================================================
DEBUG: [DOCKER:sandbox_tmp862lkupv] ERROR:
test_callbacks_can_be_added_and_removed
(test_reactive_system.TestReactiveSystem.test_callbacks_can_be_added_and
_removed)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmp862lkupv] Traceback (most ...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:50,429 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 8/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3716/107929
tokens
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:50,429 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3716/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:56,773 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    @property\n    def value(self):\n        if
self._needs_update:\n            self._cached_value =
self.compute_function([input_cell.value for input_cell in self.inputs])
\n            self._needs_update = False\n        return
self._cached_value', 'replace': '    @property\n    def value(self):\n
if self._needs_update:\n            self._cached_value =
self.compute_function([input_cell.value for input_cell in self.inputs])
\n            self._needs_update = False\n        return
self._cached_value'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:56,773 - agent -
INFO - Tool observation (first 300 chars): Error: You must get approval
before applying edits. Call get_approval_for_solution tool first....
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:56,774 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 9/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3852/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:39:56,774 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3852/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:04,095 - agent -
INFO - LLM proposed action: get_approval_for_solution with args:
{'solutions': ['Solution 1: Modify the ComputeCell to pass cell objects
to the compute function instead of values. This means changing the value
property to pass `self.inputs` directly to the compute function,
allowing the lambda functions in the tests to access `.value` on the
cell objects.\n\nSolution 2: Keep the current implementation but modify
the test expectations. However, since I cannot modify the test files,
this approach is not viable. The tests clearly expect cell objects to be
passed to the compute function.'], 'selected_solution': 1,
'reason_for_selection': 'Solution 1 is the correct approach because the
test functions are written to expect cell objects as inputs to their
lambda functions. The error messages show that the tests are trying to
access `.value` on what they expect to be cell objects. By passing the
cell objects themselves instead of their values, the compute functions
can access the `.value` property as intended in the tests.'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:04,095 - agent -
INFO - Solutions proposed. Selected index: 1. Reason: Solution 1 is the
correct approach because the test functions are written to expect cell
objects as inputs to their lambda functions. The error messages show
that the tests are trying to access `.value` on what they expect to be
cell objects. By passing the cell objects themselves instead of their
values, the compute functions can access the `.value` property as
intended in the tests.
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:04,095 - agent -
INFO - Tool observation (first 300 chars): Error: solutions must be a
list with at least 2 items....
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:04,095 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 10/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4112/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:04,096 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4112/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:11,792 - agent -
INFO - LLM proposed action: get_approval_for_solution with args:
{'solutions': ['Solution 1: Modify the ComputeCell to pass cell objects
to the compute function instead of values. This means changing the value
property to pass `self.inputs` directly to the compute function,
allowing the lambda functions in the tests to access `.value` on the
cell objects.', 'Solution 2: Create a wrapper approach where the compute
function receives cell objects but internally handles value extraction.
This would involve modifying the compute function call to maintain
backward compatibility while supporting the test expectations.'],
'selected_solution': 1, 'reason_for_selection': 'Solution 1 is the
correct approach because the test functions are written to expect cell
objects as inputs to their lambda functions. The error messages show
that the tests are trying to access `.value` on what they expect to be
cell objects. By passing the cell objects themselves instead of their
values, the compute functions can access the `.value` property as
intended in the tests.'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:11,792 - agent -
INFO - Solutions proposed. Selected index: 1. Reason: Solution 1 is the
correct approach because the test functions are written to expect cell
objects as inputs to their lambda functions. The error messages show
that the tests are trying to access `.value` on what they expect to be
cell objects. By passing the cell objects themselves instead of their
values, the compute functions can access the `.value` property as
intended in the tests.
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:11,792 - agent -
INFO - Tool observation (first 300 chars): Approved...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:11,792 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 11/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4361/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:11,793 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4361/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:22,552 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    @property\n    def value(self):\n        if
self._needs_update:\n            self._cached_value =
self.compute_function([input_cell.value for input_cell in self.inputs])
\n            self._needs_update = False\n        return
self._cached_value', 'replace': '    @property\n    def value(self):\n
if self._needs_update:\n            self._cached_value =
self.compute_function(self.inputs)\n            self._needs_update =
False\n        return self._cached_value'}
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:22,553 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:22,554 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:22,554 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 12/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4322/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:22,555 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4322/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:31,094 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_reactive_system.py']}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:31,094 - agent -
INFO - Running test command: python -m unittest test_reactive_system
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:31,155 - agent -
INFO - Tool observation (first 300 chars): FFF.F.FF.F.F
DEBUG: [DOCKER:sandbox_tmp862lkupv]
======================================================================
DEBUG: [DOCKER:sandbox_tmp862lkupv] FAIL:
test_callbacks_can_be_added_and_removed
(test_reactive_system.TestReactiveSystem.test_callbacks_can_be_added_and
_removed)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmp862lkupv] Traceback (most r...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:31,155 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 13/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3872/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:31,156 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 3872/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:38,241 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./main.py'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:38,241 - agent -
INFO - Tool observation (first 300 chars): class InputCell:
DEBUG: [DOCKER:sandbox_tmp862lkupv]     def __init__(self,
initial_value):
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.value = initial_value
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.dependents = set()
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.callbacks = set()
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]     def set_value(self, new_value):
DEBUG: [DOCKER:sandbox_tmp862lkupv]         if self.value != new_value:
DEBUG: [DOCKER:sandbox_tmp862lkupv]             self.value = new_value
DEBUG: [DOCKER:sandbox_tmp862lkupv]             # Notify all dependents
of...
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:38,241 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 14/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4113/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:38,242 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4113/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:47,890 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def _update(self):\n        old_value =
self._cached_value if not self._needs_update else None\n
self._needs_update = True\n        new_value = self.value\n        \n
# If value changed, notify dependents and callbacks\n        if
old_value != new_value:\n            for dependent in self.dependents:\n
dependent._update()\n            for callback in self.callbacks:\n
callback(new_value)', 'replace': '    def _update(self):\n        #
Store the old value before recomputing\n        old_value =
self._cached_value\n        \n        # Mark as needing update and
compute new value\n        self._needs_update = True\n        new_value
= self.value\n        \n        # If value changed, notify dependents
and callbacks\n        if old_value != new_value:\n            for
dependent in self.dependents:\n                dependent._update()\n
for callback in self.callbacks:\n                callback(new_value)'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:47,891 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:47,891 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 15/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4296/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:47,891 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4296/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:54,007 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_reactive_system.py']}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:54,007 - agent -
INFO - Running test command: python -m unittest test_reactive_system
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:54,070 - agent -
INFO - Tool observation (first 300 chars): FFF.F.FF.F.F
DEBUG: [DOCKER:sandbox_tmp862lkupv]
======================================================================
DEBUG: [DOCKER:sandbox_tmp862lkupv] FAIL:
test_callbacks_can_be_added_and_removed
(test_reactive_system.TestReactiveSystem.test_callbacks_can_be_added_and
_removed)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmp862lkupv] Traceback (most r...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:54,070 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 16/30
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4653/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:54,071 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4653/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:58,497 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./test_reactive_system.py', 'search_start_line': 18, 'search_end_line':
28}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:58,497 - agent -
INFO - Tool observation (first 300 chars): Lines 18-28 of ./
test_reactive_system.py:
DEBUG: [DOCKER:sandbox_tmp862lkupv]         compute_cell =
ComputeCell([input_cell1, input_cell2], lambda inputs: inputs[0].value +
inputs[1].value * 10)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
self.assertEqual(compute_cell.value, 21)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]     def
test_compute_cells_update_value_when_dependencies_are_changed(self):
DEBUG: [DOCKER:sandbox_tmp862lkupv]         i...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:58,499 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:58,499 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 17/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4782/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:40:58,499 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4782/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:03,462 - agent -
INFO - LLM proposed action: search_in_specified_file_v2 with args:
{'file_path': './test_reactive_system.py', 'search_term': '.value ='}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:03,466 - agent -
INFO - Tool observation (first 300 chars): (lines 21-25):
DEBUG: [DOCKER:sandbox_tmp862lkupv]     def
test_compute_cells_update_value_when_dependencies_are_changed(self):
DEBUG: [DOCKER:sandbox_tmp862lkupv]         input_cell = InputCell(1)
DEBUG: [DOCKER:sandbox_tmp862lkupv]         compute_cell =
ComputeCell([input_cell], lambda inputs: inputs[0].value + 1)
DEBUG: [DOCKER:sandbox_tmp862lkupv]         input_cell.value = 2
DEBUG: [DOCKER:sandbox_tmp862lkupv]
self.assertEqual(compute_cell.value, 3)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv] (lines 27-3...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:03,466 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 18/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4936/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:03,467 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4936/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:10,831 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': 'class InputCell:\n    def __init__(self,
initial_value):\n        self.value = initial_value\n
self.dependents = set()\n        self.callbacks = set()\n\n    def
set_value(self, new_value):\n        if self.value != new_value:\n
self.value = new_value\n            # Notify all dependents of the
change\n            for dependent in self.dependents:\n
dependent._update()', 'replace': 'class InputCell:\n    def
__init__(self, initial_value):\n        self._value = initial_value\n
self.dependents = set()\n        self.callbacks = set()\n\n
@property\n    def value(self):\n        return self._value\n\n
@value.setter\n    def value(self, new_value):\n        if self._value !
= new_value:\n            self._value = new_value\n            # Notify
all dependents of the change\n            for dependent in
self.dependents:\n                dependent._update()\n\n    def
set_value(self, new_value):\n        self.value = new_value  # Use the
property setter'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:10,831 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:10,832 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 19/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4973/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:10,832 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 4973/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:14,860 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_reactive_system.py']}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:14,860 - agent -
INFO - Running test command: python -m unittest test_reactive_system
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:14,918 - agent -
INFO - Tool observation (first 300 chars): EEE.E..E...E
DEBUG: [DOCKER:sandbox_tmp862lkupv]
======================================================================
DEBUG: [DOCKER:sandbox_tmp862lkupv] ERROR:
test_callbacks_can_be_added_and_removed
(test_reactive_system.TestReactiveSystem.test_callbacks_can_be_added_and
_removed)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmp862lkupv] Traceback (most ...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:14,919 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 20/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5368/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:14,919 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5368/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:20,151 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '            for callback in self.callbacks:\n
callback(new_value)', 'replace': '            for callback in
self.callbacks:\n                 callback(self)'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:20,151 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:20,151 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 21/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5176/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:20,152 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5176/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:25,161 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_reactive_system.py']}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:25,162 - agent -
INFO - Running test command: python -m unittest test_reactive_system
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:25,220 - agent -
INFO - Tool observation (first 300 chars): ....F.......
DEBUG: [DOCKER:sandbox_tmp862lkupv]
======================================================================
DEBUG: [DOCKER:sandbox_tmp862lkupv] FAIL:
test_callbacks_should_only_be_called_once_even_if_multiple_dependencies_
change
(test_reactive_system.TestReactiveSystem.test_callbacks_should_only_be_c
alled_once_even_if_multiple_dependencies_change)
DEBUG: [DOCKER:sandbox_tmp862lkupv] ----------...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:25,222 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:25,222 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 22/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5265/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:25,222 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5265/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:33,347 - agent -
INFO - LLM proposed action: get_approval_for_solution with args:
{'solutions': ["Solution 1: Implement a batching mechanism where
callbacks are only called after all pending updates have been processed.
This would involve tracking when updates are in progress and only
calling callbacks when the system reaches a stable state.\n\nSolution 2:
Implement a value comparison mechanism where callbacks are only called
if the final computed value is different from the value before any of
the dependency changes started. This would require storing the 'stable'
value before any updates and only calling callbacks when the final
stable value differs from that initial stable value."],
'selected_solution': 2, 'reason_for_selection': 'Solution 2 is more
appropriate for this reactive system because it ensures callbacks are
only called when the final computed value actually changes, regardless
of how many intermediate updates occurred. This matches the test
expectation where the callback should only be called once with the final
value (6) even though two input cells changed. The approach is simpler
and more deterministic than trying to batch updates.'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:33,347 - agent -
INFO - Solutions proposed. Selected index: 2. Reason: Solution 2 is more
appropriate for this reactive system because it ensures callbacks are
only called when the final computed value actually changes, regardless
of how many intermediate updates occurred. This matches the test
expectation where the callback should only be called once with the final
value (6) even though two input cells changed. The approach is simpler
and more deterministic than trying to batch updates.
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:33,347 - agent -
INFO - Tool observation (first 300 chars): Error: solutions must be a
list with at least 2 items....
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:33,347 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 23/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5227/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:33,348 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5227/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:44,331 - agent -
INFO - LLM proposed action: get_approval_for_solution with args:
{'solutions': ['Solution 1: Implement a batching mechanism where
callbacks are only called after all pending updates have been processed.
This would involve tracking when updates are in progress and only
calling callbacks when the system reaches a stable state.', "Solution 2:
Implement a value comparison mechanism where callbacks are only called
if the final computed value is different from the value before any of
the dependency changes started. This would require storing the 'stable'
value before any updates and only calling callbacks when the final
stable value differs from that initial stable value."],
'selected_solution': 2, 'reason_for_selection': 'Solution 2 is more
appropriate for this reactive system because it ensures callbacks are
only called when the final computed value actually changes, regardless
of how many intermediate updates occurred. This matches the test
expectation where the callback should only be called once with the final
value (6) even though two input cells changed. The approach is simpler
and more deterministic than trying to batch updates.'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:44,331 - agent -
INFO - Solutions proposed. Selected index: 2. Reason: Solution 2 is more
appropriate for this reactive system because it ensures callbacks are
only called when the final computed value actually changes, regardless
of how many intermediate updates occurred. This matches the test
expectation where the callback should only be called once with the final
value (6) even though two input cells changed. The approach is simpler
and more deterministic than trying to batch updates.
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:44,331 - agent -
INFO - Tool observation (first 300 chars): Approved...
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:44,331 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 24/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5510/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:44,332 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5510/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:54,597 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def _update(self):\n        # Store the old
value before recomputing\n        old_value = self._cached_value\n
\n        # Mark as needing update and compute new value\n
self._needs_update = True\n        new_value = self.value\n        \n
# If value changed, notify dependents and callbacks\n        if
old_value != new_value:\n            for dependent in self.dependents:\n
dependent._update()\n            for callback in self.callbacks:\n
callback(self)', 'replace': '    def _update(self):\n        # Store the
old value before recomputing\n        old_value = self._cached_value\n
\n        # Mark as needing update and compute new value\n
self._needs_update = True\n        new_value = self.value\n        \n
# If value changed, notify dependents first\n        if old_value !=
new_value:\n            for dependent in self.dependents:\n
dependent._update()\n            # Only call callbacks after all
dependents have been updated\n            for callback in
self.callbacks:\n                callback(self)'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:54,597 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:54,597 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 25/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5392/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:41:54,598 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5392/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:00,386 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_reactive_system.py']}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:00,386 - agent -
INFO - Running test command: python -m unittest test_reactive_system
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:00,447 - agent -
INFO - Tool observation (first 300 chars): ....F.......
DEBUG: [DOCKER:sandbox_tmp862lkupv]
======================================================================
DEBUG: [DOCKER:sandbox_tmp862lkupv] FAIL:
test_callbacks_should_only_be_called_once_even_if_multiple_dependencies_
change
(test_reactive_system.TestReactiveSystem.test_callbacks_should_only_be_c
alled_once_even_if_multiple_dependencies_change)
DEBUG: [DOCKER:sandbox_tmp862lkupv] ----------...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:00,447 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 26/30
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5542/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:00,448 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5542/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:04,136 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./test_reactive_system.py', 'search_start_line': 95, 'search_end_line':
115}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:04,136 - agent -
INFO - Tool observation (first 300 chars): Lines 95-115 of ./
test_reactive_system.py:
DEBUG: [DOCKER:sandbox_tmp862lkupv]
compute_cell.add_callback(callback1)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
compute_cell.add_callback(callback2)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
compute_cell.remove_callback(callback1)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
compute_cell.remove_callback(callback1)
DEBUG: [DOCKER:sandbox_tmp862lkupv]
compute_cell.remove_callback(callback1)
DEBUG: [DOCKER:sandbox_tmp862lkupv]         input_cell.valu...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:04,138 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:04,138 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 27/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5645/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:04,139 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5645/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:15,311 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': 'class ComputeCell:\n    def __init__(self, inputs,
compute_function):\n        self.inputs = inputs\n
self.compute_function = compute_function\n        self.dependents =
set()\n        self.callbacks = set()\n        self._needs_update =
True\n        self._cached_value = None', 'replace': 'class ComputeCell:
\n    def __init__(self, inputs, compute_function):\n        self.inputs
= inputs\n        self.compute_function = compute_function\n
self.dependents = set()\n        self.callbacks = set()\n
self._needs_update = True\n        self._cached_value = None\n
self._update_in_progress = False\n        self._pending_callbacks =
set()'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:15,311 - agent -
INFO - Tool observation (first 300 chars): Error: search string found 0
times in './main.py'. Must be found exactly once....
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:15,311 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 28/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5825/107929
tokens
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:15,312 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5825/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:18,619 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./main.py', 'search_start_line': 25, 'search_end_line': 35}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:18,619 - agent -
INFO - Tool observation (first 300 chars): Lines 25-35 of ./main.py:
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.inputs = inputs
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.compute_function =
compute_function
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.dependents = set()
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self.callbacks = set()
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self._cached_value = None
DEBUG: [DOCKER:sandbox_tmp862lkupv]         self._needs_update = True
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]         # Add this compute cell as a
dependent to all it...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:18,619 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 29/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5922/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:18,620 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 5922/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:25,087 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '        self.inputs = inputs\n
self.compute_function = compute_function\n        self.dependents =
set()\n        self.callbacks = set()\n        self._cached_value =
None\n        self._needs_update = True', 'replace': '
self.inputs = inputs\n        self.compute_function = compute_function\n
self.dependents = set()\n        self.callbacks = set()\n
self._cached_value = None\n        self._needs_update = True\n
self._update_in_progress = False\n        self._pending_callbacks =
set()'}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:25,088 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:25,088 - agent -
INFO - [RUN:8366b657-e085-4a65-b899-ef494c437792] Step 30/30
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 6029/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:25,089 - agent -
DEBUG - [REQUEST] run_id: 8366b657-e085-4a65-b899-ef494c437792, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmp862lkupv] [CONTEXT] Messages fit: 6029/107929
tokens
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:33,062 - agent -
INFO - LLM proposed action: apply_code_edit with args: {'file_path': './
main.py', 'search': '    def _update(self):\n        # Store the old
value before recomputing\n        old_value = self._cached_value\n
\n        # Mark as needing update and compute new value\n
self._needs_update = True\n        new_value = self.value\n        \n
# If value changed, notify dependents first\n        if old_value !=
new_value:\n            for dependent in self.dependents:\n
dependent._update()\n            # Only call callbacks after all
dependents have been updated\n            for callback in
self.callbacks:\n                callback(self)', 'replace': "    def
_update(self):\n        # If we're already in an update cycle, defer
callback execution\n        if self._update_in_progress:\n
return\n            \n        self._update_in_progress = True\n
\n        # Store the old value before recomputing\n        old_value =
self._cached_value\n        \n        # Mark as needing update and
compute new value\n        self._needs_update = True\n        new_value
= self.value\n        \n        # If value changed, notify dependents
first\n        if old_value != new_value:\n            for dependent in
self.dependents:\n                dependent._update()\n            #
Only call callbacks after all dependents have been updated\n
for callback in self.callbacks:\n                callback(self)\n
\n        self._update_in_progress = False"}
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:33,063 - agent -
INFO - Tool observation (first 300 chars): ok, code edit applied
successfully...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:33,063 - agent -
WARNING - Workflow ended by reaching max steps (30).
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:33,063 - agent -
INFO - Workflow execution complete. Generating final patch.
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:33,063 - agent -
INFO - Generating final git patch...
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:33,085 - agent -
INFO - Generating final git patch...
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmp862lkupv] ✅ CREATE WORKFLOW COMPLETED
DEBUG: [DOCKER:sandbox_tmp862lkupv]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:33,105 - agent -
INFO - [WORKFLOW] CREATE workflow completed
DEBUG: [DOCKER:sandbox_tmp862lkupv] HEAD is now at 382fb47 Initial
commit
DEBUG: [DOCKER:sandbox_tmp862lkupv]
DEBUG: [DOCKER:sandbox_tmp862lkupv]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmp862lkupv] 🏁 AGENT FINISHED
DEBUG: [DOCKER:sandbox_tmp862lkupv]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmp862lkupv] 2025-10-21 19:42:33,110 - agent -
INFO - [AGENT-END] Agent execution completed
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Exited agent's
agent_main()
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Writing output.json
```

```
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Wrote output.json
DEBUG: [DOCKER:sandbox_tmp862lkupv] [AGENT_RUNNER] Exiting main()
DEBUG: [SANDBOX] <sandbox_tmp862lkupv> finished running
DEBUG: [SANDBOX] <sandbox_tmp862lkupv> captured 379 lines of logs
DEBUG: [SANDBOX] Read output.json from <sandbox_tmp862lkupv>: /tmp/
tmp862lkupv/output.json
DEBUG: [PROBLEM_SUITE] Validating diff generated by
<sandbox_tmp862lkupv> for react
DEBUG: [PROBLEM_SUITE] Diff generated by <sandbox_tmp862lkupv> for react
is valid
 INFO: [PROBLEM_SUITE] Finished sandbox to run agent for problem react:
success
```

```
========== DIFF (222 lines) ==========
diff --git a/main.py b/main.py
index ab6be31..0fdd911 100644
--- a/main.py
+++ b/main.py
@@ -1,15 +1,76 @@
 class InputCell:
     def __init__(self, initial_value):
-        self.value = None
+        self._value = initial_value
+        self.dependents = set()
+        self.callbacks = set()
+
+    @property
+    def value(self):
+        return self._value
+
+    @value.setter
+    def value(self, new_value):
+        if self._value != new_value:
+            self._value = new_value
+            # Notify all dependents of the change
+            for dependent in self.dependents:
+                dependent._update()
+
+    def set_value(self, new_value):
+        self.value = new_value  # Use the property setter
 
 
 class ComputeCell:
     def __init__(self, inputs, compute_function):
-        self.value = None
+        self.inputs = inputs
+        self.compute_function = compute_function
+        self.dependents = set()
+        self.callbacks = set()
+        self._cached_value = None
+        self._needs_update = True
+        self._update_in_progress = False
+        self._pending_callbacks = set()
+
+        # Add this compute cell as a dependent to all its inputs
+        for input_cell in inputs:
+            input_cell.dependents.add(self)
+
+        # Compute initial value
+        self._update()

-    def add_callback(self, callback):
-        pass
+    @property
+    def value(self):
+        if self._needs_update:
+            self._cached_value = self.compute_function(self.inputs)
+            self._needs_update = False
+        return self._cached_value

+    def _update(self):
+        # If we're already in an update cycle, defer callback execution
+        if self._update_in_progress:
+            return
+
+        self._update_in_progress = True
+
+        # Store the old value before recomputing
+        old_value = self._cached_value
+
+        # Mark as needing update and compute new value
+        self._needs_update = True
+        new_value = self.value
+
+        # If value changed, notify dependents first
+        if old_value != new_value:
+            for dependent in self.dependents:
+                dependent._update()
+            # Only call callbacks after all dependents have been
updated
+            for callback in self.callbacks:
+                callback(self)
+
+        self._update_in_progress = False
+
+    def add_callback(self, callback):
+        self.callbacks.add(callback)
+
     def remove_callback(self, callback):
-        pass
-
\ No newline at end of file
+        self.callbacks.discard(callback)
\ No newline at end of file
diff --git a/test_reactive_system.py b/test_reactive_system.py
new file mode 100644
index 0000000..a43e584
--- /dev/null
+++ b/test_reactive_system.py
@@ -0,0 +1,126 @@
```

```
+import unittest
+from main import InputCell, ComputeCell
+
+class TestReactiveSystem(unittest.TestCase):
+
+    def test_input_cells_have_a_value(self):
+        input_cell = InputCell(10)
+        self.assertEqual(input_cell.value, 10)
+
+    def test_compute_cells_calculate_initial_value(self):
+        input_cell = InputCell(1)
+        compute_cell = ComputeCell([input_cell], lambda inputs:
inputs[0].value + 1)
+        self.assertEqual(compute_cell.value, 2)
+
+    def test_compute_cells_take_inputs_in_right_order(self):
+        input_cell1 = InputCell(1)
+        input_cell2 = InputCell(2)
+        compute_cell = ComputeCell([input_cell1, input_cell2], lambda
inputs: inputs[0].value + inputs[1].value * 10)
+        self.assertEqual(compute_cell.value, 21)
+
+    def
test_compute_cells_update_value_when_dependencies_are_changed(self):
+        input_cell = InputCell(1)
+        compute_cell = ComputeCell([input_cell], lambda inputs:
inputs[0].value + 1)
+        input_cell.value = 2
+        self.assertEqual(compute_cell.value, 3)
+
+    def test_compute_cells_can_depend_on_other_compute_cells(self):
+        input_cell = InputCell(1)
+        times_two = ComputeCell([input_cell], lambda inputs:
inputs[0].value * 2)
+        times_thirty = ComputeCell([input_cell], lambda inputs:
inputs[0].value * 30)
+        sum_cell = ComputeCell([times_two, times_thirty], lambda
inputs: inputs[0].value + inputs[1].value)
+        self.assertEqual(sum_cell.value, 32)
+        input_cell.value = 3
+        self.assertEqual(sum_cell.value, 96)
+
+    def test_compute_cells_fire_callbacks(self):
+        input_cell = InputCell(1)
+        output = []
+        callback = lambda cell: output.append(cell.value)
+        compute_cell = ComputeCell([input_cell], lambda inputs:
inputs[0].value + 1)
+        compute_cell.add_callback(callback)
+        input_cell.value = 2
+        self.assertEqual(output, [3])
+
+    def test_callbacks_only_fire_on_change(self):
+        input_cell = InputCell(1)
+        output = []
+        callback = lambda cell: output.append(cell.value)
+        compute_cell = ComputeCell([input_cell], lambda inputs: 111 if
inputs[0].value < 3 else 222)
+        compute_cell.add_callback(callback)
+        input_cell.value = 2
+        self.assertEqual(output, [])
+        input_cell.value = 4
+        self.assertEqual(output, [222])
+
+    def test_callbacks_do_not_report_already_reported_values(self):
+        input_cell = InputCell(1)
+        output = []
+        callback = lambda cell: output.append(cell.value)
+        compute_cell = ComputeCell([input_cell], lambda inputs:
inputs[0].value + 1)
+        compute_cell.add_callback(callback)
+        input_cell.value = 2
+        self.assertEqual(output, [3])
+        input_cell.value = 3
+        self.assertEqual(output, [3, 4])
+
+    def test_callbacks_can_be_added_and_removed(self):
+        input_cell = InputCell(1)
+        output1 = []
+        output2 = []
+        output3 = []
+        callback1 = lambda cell: output1.append(cell.value)
+        callback2 = lambda cell: output2.append(cell.value)
+        callback3 = lambda cell: output3.append(cell.value)
+        compute_cell = ComputeCell([input_cell], lambda inputs:
inputs[0].value + 1)
+        compute_cell.add_callback(callback1)
+        compute_cell.add_callback(callback2)
+        input_cell.value = 2
+        self.assertEqual(output1, [3])
+        self.assertEqual(output2, [3])
+        compute_cell.remove_callback(callback1)
+        compute_cell.add_callback(callback3)
+        input_cell.value = 3
+        self.assertEqual(output1, [3])
+        self.assertEqual(output2, [3, 4])
+        self.assertEqual(output3, [4])
+
+    def
test_removing_a_callback_multiple_times_doesn_t_interfere_with_other_cal
lbacks(self):
+        input_cell = InputCell(1)
+        output1 = []
+        output2 = []
+        callback1 = lambda cell: output1.append(cell.value)
+        callback2 = lambda cell: output2.append(cell.value)
+        compute_cell = ComputeCell([input_cell], lambda inputs:
inputs[0].value + 1)
+        compute_cell.add_callback(callback1)
+        compute_cell.add_callback(callback2)
+        compute_cell.remove_callback(callback1)
+        compute_cell.remove_callback(callback1)
+        compute_cell.remove_callback(callback1)
+        input_cell.value = 2
+        self.assertEqual(output1, [])
+        self.assertEqual(output2, [3])
+
+    def
test_callbacks_should_only_be_called_once_even_if_multiple_dependencies_
change(self):
+        input_cell1 = InputCell(1)
+        input_cell2 = InputCell(2)
+        output = []
+        callback = lambda cell: output.append(cell.value)
+        compute_cell = ComputeCell([input_cell1, input_cell2], lambda
inputs: inputs[0].value + inputs[1].value)
+        compute_cell.add_callback(callback)
+        input_cell1.value = 2
+        input_cell2.value = 4
+        self.assertEqual(output, [6])
+
+    def
test_callbacks_should_not_be_called_if_dependencies_change_but_output_va
lue_doesn_t_change(self):
+        input_cell1 = InputCell(1)
+        input_cell2 = InputCell(2)
+        output = []
+        callback = lambda cell: output.append(cell.value)
+        compute_cell = ComputeCell([input_cell1, input_cell2], lambda
inputs: 7)
+        compute_cell.add_callback(callback)
+        input_cell1.value = 2
+        self.assertEqual(output, [])
+
+if __name__ == '__main__':
+    unittest.main()
\ No newline at end of file

========== LOGS (379 lines) ==========
```

```
 INFO: [PROBLEM_SUITE] Starting sandbox to evaluate solution diff for
problem react
DEBUG: [SANDBOX] Created sandbox temp directory for
<sandbox_tmpxuqjwucy>: /tmp/tmpxuqjwucy
DEBUG: [POLYGLOT] Copied main.py to /tmp/tmpxuqjwucy/repo for react
DEBUG: [POLYGLOT] Copied tests.py to /tmp/tmpxuqjwucy/repo for react
DEBUG: [POLYGLOT] Initializing git repository in /tmp/tmpxuqjwucy/repo
for react
DEBUG: [GIT] Initializing git repository in /tmp/tmpxuqjwucy/repo
DEBUG: [GIT] Initialized git repository in /tmp/tmpxuqjwucy/repo
DEBUG: [GIT] Adding all files in /tmp/tmpxuqjwucy/repo
DEBUG: [GIT] Added all files in /tmp/tmpxuqjwucy/repo
DEBUG: [GIT] Making initial commit: Initial commit
DEBUG: [GIT] Made initial commit: Initial commit
DEBUG: [POLYGLOT] Initialized git repository in /tmp/tmpxuqjwucy/repo
for react
```

```
DEBUG: [PROBLEM_SUITE] Applying agent's solution diff to /tmp/
tmpxuqjwucy/repo for problem react
DEBUG: [PROBLEM_SUITE] Applied agent's solution diff to /tmp/
tmpxuqjwucy/repo for problem react
DEBUG: [SANDBOX] Copied main Python script (/root/abstract-agent-runner/
problem_suites/polyglot/TEST_RUNNER.py) for <sandbox_tmpxuqjwucy>: /tmp/
tmpxuqjwucy/TEST_RUNNER.py
DEBUG: [SANDBOX] Written input.json for <sandbox_tmpxuqjwucy>: /tmp/
tmpxuqjwucy/input.json
DEBUG: [SANDBOX] Running sandbox <sandbox_tmpxuqjwucy>
DEBUG: [SANDBOX] Started sandbox runner thread for <sandbox_tmpxuqjwucy>
DEBUG: [PROBLEM_SUITE] Started sandbox to evaluate solution diff for
problem react
DEBUG: [SANDBOX] Cleaned up sandbox <sandbox_tmp862lkupv>
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Entered
main()
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Loading
main.py
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Loaded
main.py
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Loading
tests.py
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Loaded
tests.py
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Found test
class: ReactTest
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Found 14 test
methods
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] [1/14]
Running test_an_input_cell_s_value_can_be_set...
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER]
test_an_input_cell_s_value_can_be_set: PASSED
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] [2/14]
Running test_callback_cells_only_fire_on_change...
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER]
test_callback_cells_only_fire_on_change: FAILED - '<' not supported
between instances of 'InputCell' and 'int'
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Test results:
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [{'name':
'test_an_input_cell_s_value_can_be_set', 'status': 'pass'}, {'name':
'test_callback_cells_only_fire_on_change', 'status': 'fail'}, {'name':
'test_callbacks_can_be_added_and_removed', 'status': 'skip'}, {'name':
'test_callbacks_can_fire_from_multiple_cells', 'status': 'skip'},
{'name': 'test_callbacks_do_not_report_already_reported_values',
'status': 'skip'}, {'name':
'test_callbacks_should_not_be_called_if_dependencies_change_but_output_v
alue_doesn_t_change', 'status': 'skip'}, {'name':
'test_callbacks_should_only_be_called_once_even_if_multiple_dependencies
_change', 'status': 'skip'}, {'name':
'test_compute_cells_calculate_initial_value', 'status': 'skip'},
{'name': 'test_compute_cells_can_depend_on_other_compute_cells',
'status': 'skip'}, {'name': 'test_compute_cells_fire_callbacks',
'status': 'skip'}, {'name':
'test_compute_cells_take_inputs_in_the_right_order', 'status': 'skip'},
{'name':
'test_compute_cells_update_value_when_dependencies_are_changed',
'status': 'skip'}, {'name': 'test_input_cells_have_a_value', 'status':
'skip'}, {'name':
'test_removing_a_callback_multiple_times_doesn_t_interfere_with_other_ca
llbacks', 'status': 'skip'}]
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Test summary:
1 passed, 1 failed, 12 skipped
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Writing
output.json
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Wrote
output.json
DEBUG: [DOCKER:sandbox_tmpxuqjwucy] [POLYGLOT_TEST_RUNNER] Exiting
main()
DEBUG: [SANDBOX] <sandbox_tmpxuqjwucy> finished running
DEBUG: [SANDBOX] <sandbox_tmpxuqjwucy> captured 17 lines of logs
DEBUG: [SANDBOX] Read output.json from <sandbox_tmpxuqjwucy>: /tmp/
tmpxuqjwucy/output.json
 INFO: [PROBLEM_SUITE] Finished sandbox to evaluate solution diff for
problem react: success



========== TEST RESULTS ==========
1 passed, 1 failed, 12 skipped
test_an_input_cell_s_value_can_be_set - no category - pass
test_callback_cells_only_fire_on_change - no category - fail
test_callbacks_can_be_added_and_removed - no category - skip
test_callbacks_can_fire_from_multiple_cells - no category - skip
test_callbacks_do_not_report_already_reported_values - no category -
skip
test_callbacks_should_not_be_called_if_dependencies_change_but_output_va
lue_doesn_t_change - no category - skip
test_callbacks_should_only_be_called_once_even_if_multiple_dependencies_
change - no category - skip
test_compute_cells_calculate_initial_value - no category - skip
test_compute_cells_can_depend_on_other_compute_cells - no category -
skip
test_compute_cells_fire_callbacks - no category - skip
test_compute_cells_take_inputs_in_the_right_order - no category - skip
test_compute_cells_update_value_when_dependencies_are_changed - no
category - skip
test_input_cells_have_a_value - no category - skip
test_removing_a_callback_multiple_times_doesn_t_interfere_with_other_cal
lbacks - no category - skip
========== LOGS (17 lines) ==========



DEBUG: [SANDBOX] Cleaned up sandbox <sandbox_tmpxuqjwucy>
```

```
(.venv_clean) root@kind-name-wilts-fin-01:~/abstract-agent-runner#
python cli.py polyglot proverb  agent-maxi-2.py http://
135.181.71.13:8000 --log-docker-to-stdout --verbose --timeout 1200
 INFO: [POLYGLOT] Loaded 33 problems from datasets/polyglot/
polyglot.json
DEBUG: [POLYGLOT]     Problem affine-cipher verified successfully (found
16 associated tests)
DEBUG: [POLYGLOT]     Problem beer-song verified successfully (found 8
associated tests)
DEBUG: [POLYGLOT]     Problem book-store verified successfully (found 20
associated tests)
DEBUG: [POLYGLOT]     Problem bottle-song verified successfully (found 7
associated tests)
DEBUG: [POLYGLOT]     Problem bowling verified successfully (found 31
associated tests)
DEBUG: [POLYGLOT]     Problem connect verified successfully (found 10
associated tests)
DEBUG: [POLYGLOT]     Problem dominoes verified successfully (found 13
associated tests)
DEBUG: [POLYGLOT]     Problem dot-dsl verified successfully (found 12
associated tests)
DEBUG: [POLYGLOT]     Problem food-chain verified successfully (found 10
associated tests)
DEBUG: [POLYGLOT]     Problem forth verified successfully (found 54
associated tests)
DEBUG: [POLYGLOT]     Problem go-counting verified successfully (found
11 associated tests)
DEBUG: [POLYGLOT]     Problem grade-school verified successfully (found
20 associated tests)
DEBUG: [POLYGLOT]     Problem grep verified successfully (found 25
associated tests)
DEBUG: [POLYGLOT]     Problem hangman verified successfully (found 7
associated tests)
DEBUG: [POLYGLOT]     Problem list-ops verified successfully (found 24
associated tests)
DEBUG: [POLYGLOT]     Problem phone-number verified successfully (found
21 associated tests)
DEBUG: [POLYGLOT]     Problem pig-latin verified successfully (found 22
associated tests)
DEBUG: [POLYGLOT]     Problem poker verified successfully (found 37
associated tests)
DEBUG: [POLYGLOT]     Problem pov verified successfully (found 15
associated tests)
DEBUG: [POLYGLOT]     Problem proverb verified successfully (found 8
associated tests)
DEBUG: [POLYGLOT]     Problem react verified successfully (found 14
associated tests)
DEBUG: [POLYGLOT]     Problem rest-api verified successfully (found 9
associated tests)
DEBUG: [POLYGLOT]     Problem robot-name verified successfully (found 4
associated tests)
DEBUG: [POLYGLOT]     Problem scale-generator verified successfully
(found 17 associated tests)
```

```
DEBUG: [POLYGLOT]     Problem sgf-parsing verified successfully (found
23 associated tests)
DEBUG: [POLYGLOT]     Problem simple-linked-list verified successfully
(found 20 associated tests)
DEBUG: [POLYGLOT]     Problem transpose verified successfully (found 12
associated tests)
DEBUG: [POLYGLOT]     Problem tree-building verified successfully (found
13 associated tests)
DEBUG: [POLYGLOT]     Problem two-bucket verified successfully (found 9
associated tests)
DEBUG: [POLYGLOT]     Problem variable-length-quantity verified
successfully (found 26 associated tests)
DEBUG: [POLYGLOT]     Problem wordy verified successfully (found 25
associated tests)
DEBUG: [POLYGLOT]     Problem zebra-puzzle verified successfully (found
2 associated tests)
DEBUG: [POLYGLOT]     Problem zipper verified successfully (found 14
associated tests)
 INFO: [POLYGLOT] Successfully loaded 33 problems
 INFO: Problem proverb has 8 tests
 INFO: [SANDBOX] Checking gateway URL: http://135.181.71.13:8000
 INFO: [SANDBOX] Gateway URL http://135.181.71.13:8000 is valid
DEBUG: [SANDBOX] Stopping and deleting all containers
DEBUG: [SANDBOX] Stopped and deleted all containers
 INFO: [SANDBOX] Building Docker image: sandbox-image
[+] Building 0.8s (11/11) FINISHED
docker:default
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 1.04kB
0.0s
 => [internal] load metadata for docker.io/library/python:3.11-slim
0.7s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [1/6] FROM docker.io/library/python:3.11-
slim@sha256:b6000fc45f769f42c4c717dab2675bbb0ec6531c32a0483a2f78de0b7023
e71b
0.0s
 => => resolve docker.io/library/python:3.11-
slim@sha256:b6000fc45f769f42c4c717dab2675bbb0ec6531c32a0483a2f78de0b7023
e71b
0.0s
 => [internal] load build context
0.0s
 => => transferring context: 46B
0.0s
 => CACHED [2/6] RUN apt-get update &&     apt-get install -y --no-
install-recommends git patch diffutils &&     rm -rf /var/lib/apt/lists/
*
0.0s
 => CACHED [3/6] COPY sandbox_requirements.txt /tmp/
sandbox_requirements.txt
0.0s
```

```
 => CACHED [4/6] RUN pip install --no-cache-dir --upgrade pip &&     pip
install --no-cache-dir -r /tmp/sandbox_requirements.txt &&     rm /tmp/
sandbox_requirements.txt                                         0.0s
 => CACHED [5/6] RUN pip cache purge &&     rm -rf /root/.cache/pip
0.0s
 => CACHED [6/6] WORKDIR /sandbox
0.0s
 => exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:6a0d37b7a80c209efc720b51333a5af41822e44e249d5749ba8c3dcb2762dbcd
0.0s
 => => naming to docker.io/library/sandbox-image
0.0s
 INFO: [SANDBOX] Successfully built Docker image: sandbox-image
DEBUG: [SANDBOX] Found sandbox network: sandbox-network
 INFO: [SANDBOX] Building Docker image: sandbox-proxy-image
[+] Building 0.8s (8/8) FINISHED
docker:default
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 289B
0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine
0.7s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [1/3] FROM docker.io/library/
nginx:alpine@sha256:61e01287e546aac28a3f56839c136b31f590273f3b41187a36f4
6f6a03bbfe22
0.0s
 => [internal] load build context
0.0s
 => => transferring context: 41B
0.0s
 => CACHED [2/3] RUN apk add --no-cache gettext
0.0s
 => CACHED [3/3] COPY nginx.conf.template /tmp/nginx.conf.template
0.0s
 => exporting to image
0.0s
 => => exporting layers
0.0s
 => => writing image
sha256:b38edefd62e5475e622d31caf7dabe385f28d175b050c8ae714b02c9758742a3
0.0s
 => => naming to docker.io/library/sandbox-proxy-image
0.0s
 INFO: [SANDBOX] Successfully built Docker image: sandbox-proxy-image
 INFO: [SANDBOX] Running sandbox proxy
DEBUG: [SANDBOX] Connected sandbox proxy to bridge network
DEBUG: [SANDBOX] Starting watchdog thread
```

```
DEBUG: [SANDBOX] Started watchdog thread
 INFO: [PROBLEM_SUITE] Starting sandbox to run agent for problem proverb
DEBUG: [SANDBOX] Created sandbox temp directory for
<sandbox_tmpk95tjvac>: /tmp/tmpk95tjvac
DEBUG: [POLYGLOT] Copied main.py to /tmp/tmpk95tjvac/repo for proverb
DEBUG: [POLYGLOT] Initializing git repository in /tmp/tmpk95tjvac/repo
for proverb
DEBUG: [GIT] Initializing git repository in /tmp/tmpk95tjvac/repo
DEBUG: [GIT] Initialized git repository in /tmp/tmpk95tjvac/repo
DEBUG: [GIT] Adding all files in /tmp/tmpk95tjvac/repo
DEBUG: [GIT] Added all files in /tmp/tmpk95tjvac/repo
DEBUG: [GIT] Making initial commit: Initial commit
DEBUG: [GIT] Made initial commit: Initial commit
DEBUG: [POLYGLOT] Initialized git repository in /tmp/tmpk95tjvac/repo
for proverb
DEBUG: [SANDBOX] Copied main Python script (/root/abstract-agent-runner/
problem_suites/AGENT_RUNNER.py) for <sandbox_tmpk95tjvac>: /tmp/
tmpk95tjvac/AGENT_RUNNER.py
DEBUG: [SANDBOX] Written input.json for <sandbox_tmpk95tjvac>: /tmp/
tmpk95tjvac/input.json
DEBUG: [SANDBOX] Running sandbox <sandbox_tmpk95tjvac>
DEBUG: [SANDBOX] Started sandbox runner thread for <sandbox_tmpk95tjvac>
DEBUG: [PROBLEM_SUITE] Started sandbox to run agent for problem proverb
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Entered main()
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Reading input.json
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Read input.json
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Loading /sandbox/
agent.py
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Loaded /sandbox/
agent.py
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] agent_main() function
found in /sandbox/agent.py
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Entering agent's
agent_main()
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 🚀 AGENT STARTING
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:23,857 - agent -
INFO - [AGENT-START] Run ID: a8f13f5e-b73c-4562-9cfe-7c575bd476ba
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:23,857 - agent -
INFO - [AGENT-START] Repo directory: /sandbox/repo
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:23,857 - agent -
INFO - [AGENT-START] Test mode: False
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:23,857 - agent -
INFO - [AGENT-START] Timeout: 1800s, Max steps: 400
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [DEBUG] Starting git initialization
check...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [DEBUG] Work directory: /sandbox/
repo
```

```
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [DEBUG] Before chdir - pwd shows: /
sandbox/repo
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [DEBUG] After chdir - pwd shows: /
sandbox/repo
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [DEBUG] Git repository already
exists
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:23,865 - agent -
INFO - [AGENT-START] Problem statement length: 1413 chars
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
────────────────────────────────────────────────────────────────────────
────────

DEBUG: [DOCKER:sandbox_tmpk95tjvac] 🔍 DETERMINING PROBLEM TYPE...
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
────────────────────────────────────────────────────────────────────────
────────

DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:23,865 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 279/219340
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,900 - agent -
INFO - [PROBLEM-TYPE] Determined: CREATE
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ✅ Problem type: CREATE
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
────────────────────────────────────────────────────────────────────────
────────

DEBUG: [DOCKER:sandbox_tmpk95tjvac] 🆕 CREATE TASK DETECTED - STARTING
CREATE WORKFLOW
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
────────────────────────────────────────────────────────────────────────
────────

DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,900 - agent -
INFO - [WORKFLOW] Starting CREATE task workflow
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
┌───────────────────────────────────────────────────────────────────────
───────┐
DEBUG: [DOCKER:sandbox_tmpk95tjvac] │ 🆕 CREATE TASK WORKFLOW STARTING
│
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
└───────────────────────────────────────────────────────────────────────
───────┘
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,900 - agent -
INFO - [CREATE] Starting CREATE task workflow
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,900 - agent -
INFO - [CREATE] Problem statement length: 1465 chars
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,900 - agent -
INFO - [CREATE] Step 1: Generating code skeleton
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,901 - agent -
INFO - [CREATE] Code skeleton generated: 38 chars
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,901 - agent -
INFO - [CREATE] Step 2: Generating initial solution
```

```
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,901 - agent -
INFO - Starting multi-step reasoning solution generation
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:25,901 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 706/219340
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:27,445 - agent -
INFO - Multi-step reasoning solution generation completed successfully
with infinite loop validation
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:27,445 - agent -
INFO - Generated initial solution successfully using multi-step
reasoning
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:27,445 - agent -
INFO - [CREATE] Initial solution generated: 399 chars
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:27,445 - agent -
INFO - [CREATE] Step 3: Extracting and writing solution files
DEBUG: [DOCKER:sandbox_tmpk95tjvac] Created file: ./main.py
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:27,446 - agent -
INFO - [CREATE] Created 1 solution files
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ✅ Created 1 solution files
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:27,446 - agent -
INFO - [CREATE] Step 4: Generating test files
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:27,446 - agent -
INFO - Starting test cases generation
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:27,446 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 420/219340
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:31,324 - agent -
INFO - Step 1 - Testcase Generation completed
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:31,324 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
Qwen/Qwen3-Coder-480B-A35B-Instruct-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 544/219340
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,719 - agent -
INFO - Step 2 - Testcase check completed
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,719 - agent -
INFO - Multi-step reasoning solution generation completed successfully
with infinite loop validation
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,719 - agent -
INFO - Generated testcases successfully using multi-step reasoning
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,719 - agent -
INFO - [CREATE] Test cases generated: 1737 chars
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,719 - agent -
INFO - [CREATE] Step 5: Extracting and writing test files
DEBUG: [DOCKER:sandbox_tmpk95tjvac] Created file: ./test_proverb.py
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - [CREATE] Created 1 test files
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ✅ Created 1 test files
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - [CREATE] Time spent: 17.8s, Remaining timeout: 1722.2s
```

```
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - [CREATE] Step 6: Starting iterative FIX workflow for refinement
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 🔧 Starting iterative refinement
workflow...
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
┌───────────────────────────────────────────────────────────────────────
───────┐
DEBUG: [DOCKER:sandbox_tmpk95tjvac] │ 🔧 ITERATIVE FIX WORKFLOW STARTING
│
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
└───────────────────────────────────────────────────────────────────────
───────┘
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - [ITERATIVE] Starting iterative FIX workflow
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - [ITERATIVE] Run ID: a8f13f5e-b73c-4562-9cfe-7c575bd476ba
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - [ITERATIVE] Max steps: 30, Timeout: 1722.1805226802826s
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - [ITERATIVE] Test runner: unittest (mode: FILE)
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - State file not found at .agent_state.json. Starting with a fresh
state.
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,720 - agent -
INFO - [ITERATIVE] Loaded COT state: 0 existing thoughts
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,722 - agent -
INFO - [ITERATIVE] Initialized tool manager with 13 tools
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,722 - agent -
INFO - [ITERATIVE] Workflow started at 19:48:43
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 🚀 Starting iterative workflow (max
30 steps, 1722.1805226802826s timeout)
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,722 - agent -
INFO - [RUN:a8f13f5e-b73c-4562-9cfe-7c575bd476ba] Step 1/30
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 2474/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:43,723 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 2474/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:52,617 - agent -
INFO - LLM proposed action: search_in_all_files_content with args:
{'search_term': 'proverb'}
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:52,618 - agent -
INFO - Tool observation (first 300 chars): ./main.py:1:def
proverb(*args, qualifier=None):
```

```
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ./test_proverb.py:2:from main import
proverb
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ./test_proverb.py:5:class
TestProverb(unittest.TestCase):
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ./
test_proverb.py:7:self.assertEqual(proverb(), [])
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ./
test_proverb.py:10:self.assertEqual(proverb("nail"), ["And all for the
want of a nail."])
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ./tes...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:52,618 - agent -
INFO - [RUN:a8f13f5e-b73c-4562-9cfe-7c575bd476ba] Step 2/30
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 2598/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:52,618 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 2598/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:58,181 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./main.py'}
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:58,181 - agent -
INFO - Tool observation (first 300 chars): def proverb(*args,
qualifier=None):
DEBUG: [DOCKER:sandbox_tmpk95tjvac]     if not args:
DEBUG: [DOCKER:sandbox_tmpk95tjvac]         return []
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]     lines = []
DEBUG: [DOCKER:sandbox_tmpk95tjvac]     for i in range(len(args) - 1):
DEBUG: [DOCKER:sandbox_tmpk95tjvac]         lines.append(f"For want of a
{args[i]} the {args[i+1]} was lost.")
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]     if qualifier:
DEBUG: [DOCKER:sandbox_tmpk95tjvac]         lines.append(f"And all for
the want of a {qualifier} {args[0]}.")
DEBUG: [DOCKER:sandbox_tmpk95tjvac]   ...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:58,181 - agent -
INFO - [RUN:a8f13f5e-b73c-4562-9cfe-7c575bd476ba] Step 3/30
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 2697/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:48:58,181 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 2697/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:01,292 - agent -
INFO - LLM proposed action: get_file_content with args: {'file_path':
'./test_proverb.py'}
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:01,292 - agent -
INFO - Tool observation (first 300 chars): import unittest
DEBUG: [DOCKER:sandbox_tmpk95tjvac] from main import proverb
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac] class
TestProverb(unittest.TestCase):
```

```
DEBUG: [DOCKER:sandbox_tmpk95tjvac]     def test_empty_list(self):
DEBUG: [DOCKER:sandbox_tmpk95tjvac]         self.assertEqual(proverb(),
[])
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]     def test_single_item(self):
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
self.assertEqual(proverb("nail"), ["And all for the want of a nail."])
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]     def test_two_items(self):
DEBUG: [DOCKER:sandbox_tmpk95tjvac]      ...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:01,292 - agent -
INFO - [RUN:a8f13f5e-b73c-4562-9cfe-7c575bd476ba] Step 4/30
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 2908/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:01,293 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 2908/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:08,734 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_proverb.py']}
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:08,734 - agent -
INFO - Running test command: python -m unittest test_proverb
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:08,836 - agent -
INFO - Tool observation (first 300 chars): .......
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmpk95tjvac] Ran 7 tests in 0.000s
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac] OK
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:08,836 - agent -
INFO - [RUN:a8f13f5e-b73c-4562-9cfe-7c575bd476ba] Step 5/30
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 3039/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:08,837 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 3039/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:14,404 - agent -
INFO - LLM proposed action: run_repo_tests with args: {'file_paths':
['./test_proverb.py']}
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:14,404 - agent -
INFO - Running test command: python -m unittest test_proverb
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:14,460 - agent -
INFO - Tool observation (first 300 chars): .......
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
----------------------------------------------------------------------
DEBUG: [DOCKER:sandbox_tmpk95tjvac] Ran 7 tests in 0.000s
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac] OK
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:14,460 - agent -
INFO - [RUN:a8f13f5e-b73c-4562-9cfe-7c575bd476ba] Step 6/30
```

```
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 3129/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:14,461 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 3129/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:25,929 - agent -
INFO - LLM proposed action: run_code with args: {'content': 'from main
import proverb\n\n# Test the example from the problem statement\nresult
= proverb("nail", "shoe", "horse", "rider", "message", "battle",
"kingdom")\nprint("Expected output:")\nprint([\n    "For want of a nail
the shoe was lost.",\n    "For want of a shoe the horse was lost.",\n
"For want of a horse the rider was lost.",\n    "For want of a rider the
message was lost.",\n    "For want of a message the battle was lost.",\n
"For want of a battle the kingdom was lost.",\n    "And all for the want
of a nail."\n])\nprint("\\nActual output:")\nprint(result)\n\n# Test
with qualifier\nresult_with_qualifier = proverb("nail", "shoe",
qualifier="horseshoe")\nprint("\\nWith qualifier expected:")\nprint([\n
"For want of a nail the shoe was lost.",\n    "And all for the want of a
horseshoe nail."\n])\nprint("\\nWith qualifier actual:")
\nprint(result_with_qualifier)', 'file_path': './test_verification.py'}
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:25,945 - agent -
INFO - Tool observation (first 300 chars): Expected output:
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ['For want of a nail the shoe was
lost.', 'For want of a shoe the horse was lost.', 'For want of a horse
the rider was lost.', 'For want of a rider the message was lost.', 'For
want of a message the battle was lost.', 'For want of a battle the
kingdom was lost.', 'And all for the wa...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:25,946 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:25,946 - agent -
INFO - [RUN:a8f13f5e-b73c-4562-9cfe-7c575bd476ba] Step 7/30
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 3366/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:25,947 - agent -
DEBUG - [REQUEST] run_id: a8f13f5e-b73c-4562-9cfe-7c575bd476ba, model:
zai-org/GLM-4.5-FP8
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [CONTEXT] Messages fit: 3366/107929
tokens
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,325 - agent -
INFO - LLM proposed action: finish with args: {'investigation_summary':
"After thorough investigation, I found that the current implementation
in `main.py` is already correct and complete. The `proverb` function
properly handles all test cases:\n\n1. **Empty list**: Returns an empty
list as expected\n2. **Single item**: Returns only the final verse
referencing that item\n3. **Multiple items**: Generates the complete
chain of 'For want of a [item1] the [item2] was lost.' lines\n4.
**Qualifier parameter**: Correctly modifies the final verse by placing
the qualifier before the first item\n\nThe implementation uses proper
unpacking with `*args` to handle lists of arbitrary length and correctly
processes the `qualifier` keyword argument. All 7 test cases pass
successfully, and manual verification confirms the output matches the
expected format from the problem statement exactly. No changes are
needed as the code already meets all requirements."}
```

```
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,325 - agent -
INFO - Finish called. Summary: After thorough investigation, I found
that the current implementation in `main.py` is already correct and
complete. The `proverb` function properly handles all test cases:
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 1. **Empty list**: Returns an empty
list as expected
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2. **Single item**: Returns only the
final verse referencing that item
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 3. **Multiple items**: Generates the
complete chain of 'For want of a [item1] the [item2] was lost.' lines
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 4. **Qualifier parameter**:
Correctly modifies the final verse by placing the qualifier before the
first item
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac] The implementation uses proper
unpacking with `*args` to handle lists of arbitrary length and correctly
processes the `qualifier` keyword argument. All 7 test cases pass
successfully, and manual verification confirms the output matches the
expected format from the problem statement exactly. No changes are
needed as the code already meets all requirements.
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,325 - agent -
INFO - Tool observation (first 300 chars): finish...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,326 - agent -
DEBUG - Successfully saved agent state to .agent_state.json
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,326 - agent -
INFO - Workflow finished successfully via "finish" tool.
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,326 - agent -
INFO - Workflow execution complete. Generating final patch.
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,326 - agent -
INFO - Generating final git patch...
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,348 - agent -
INFO - Generating final git patch...
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmpk95tjvac] ✅ CREATE WORKFLOW COMPLETED
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
────────────────────────────────────────────────────────────────────────
────────
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,367 - agent -
INFO - [WORKFLOW] CREATE workflow completed
DEBUG: [DOCKER:sandbox_tmpk95tjvac] HEAD is now at b9f2ea5 Initial
commit
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 🏁 AGENT FINISHED
DEBUG: [DOCKER:sandbox_tmpk95tjvac]
========================================================================
========
DEBUG: [DOCKER:sandbox_tmpk95tjvac] 2025-10-21 19:49:40,372 - agent -
INFO - [AGENT-END] Agent execution completed
```

```
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Exited agent's
agent_main()
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Writing output.json
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Wrote output.json
DEBUG: [DOCKER:sandbox_tmpk95tjvac] [AGENT_RUNNER] Exiting main()
DEBUG: [SANDBOX] <sandbox_tmpk95tjvac> finished running
DEBUG: [SANDBOX] <sandbox_tmpk95tjvac> captured 198 lines of logs
DEBUG: [SANDBOX] Read output.json from <sandbox_tmpk95tjvac>: /tmp/
tmpk95tjvac/output.json
DEBUG: [PROBLEM_SUITE] Validating diff generated by
<sandbox_tmpk95tjvac> for proverb
DEBUG: [PROBLEM_SUITE] Diff generated by <sandbox_tmpk95tjvac> for
proverb is valid
 INFO: [PROBLEM_SUITE] Finished sandbox to run agent for problem
proverb: success
```

```
========== DIFF (117 lines) ==========
diff --git a/main.py b/main.py
index d1be410..1c31f20 100644
--- a/main.py
+++ b/main.py
@@ -1,2 +1,14 @@
-def proverb():
-    pass
+def proverb(*args, qualifier=None):
+    if not args:
+        return []
+
+    lines = []
+    for i in range(len(args) - 1):
+        lines.append(f"For want of a {args[i]} the {args[i+1]} was
lost.")
+
+    if qualifier:
+        lines.append(f"And all for the want of a {qualifier}
{args[0]}.")
+    else:
+        lines.append(f"And all for the want of a {args[0]}.")
+
+    return lines
\ No newline at end of file
diff --git a/test_proverb.py b/test_proverb.py
new file mode 100644
index 0000000..3ffa8ed
--- /dev/null
+++ b/test_proverb.py
@@ -0,0 +1,55 @@
+import unittest
+from main import proverb
+
+
+class TestProverb(unittest.TestCase):
+    def test_empty_list(self):
+        self.assertEqual(proverb(), [])
```

```
+
+    def test_single_item(self):
+        self.assertEqual(proverb("nail"), ["And all for the want of a 
nail."])
+
+    def test_two_items(self):
+        self.assertEqual(
+            proverb("nail", "shoe"),
+            [
+                "For want of a nail the shoe was lost.",
+                "And all for the want of a nail."
+            ]
+        )
+
+    def test_multiple_items(self):
+        self.assertEqual(
+            proverb("nail", "shoe", "horse", "rider", "message", 
"battle", "kingdom"),
+            [
+                "For want of a nail the shoe was lost.",
+                "For want of a shoe the horse was lost.",
+                "For want of a horse the rider was lost.",
+                "For want of a rider the message was lost.",
+                "For want of a message the battle was lost.",
+                "For want of a battle the kingdom was lost.",
+                "And all for the want of a nail."
+            ]
+        )
+
+    def test_qualifier(self):
+        self.assertEqual(
+            proverb("nail", "shoe", qualifier="horseshoe"),
+            [
+                "For want of a nail the shoe was lost.",
+                "And all for the want of a horseshoe nail."
+            ]
+        )
+
+    def test_qualifier_with_single_item(self):
+        self.assertEqual(
+            proverb("nail", qualifier="horseshoe"),
+            ["And all for the want of a horseshoe nail."]
+        )
+
+    def test_qualifier_with_empty_list(self):
+        self.assertEqual(proverb(qualifier="horseshoe"), [])
+
+
+if __name__ == "__main__":
+    unittest.main()
\ No newline at end of file
diff --git a/test_verification.py b/test_verification.py
new file mode 100644
index 0000000..9238084
--- /dev/null
+++ b/test_verification.py
```

```
@@ -0,0 +1,26 @@
+from main import proverb
+
+# Test the example from the problem statement
+result = proverb("nail", "shoe", "horse", "rider", "message", "battle", "kingdom")
+print("Expected output:")
+print([
+    "For want of a nail the shoe was lost.",
+    "For want of a shoe the horse was lost.",
+    "For want of a horse the rider was lost.",
+    "For want of a rider the message was lost.",
+    "For want of a message the battle was lost.",
+    "For want of a battle the kingdom was lost.",
+    "And all for the want of a nail."
+])
+print("\nActual output:")
+print(result)
+
+# Test with qualifier
+result_with_qualifier = proverb("nail", "shoe", qualifier="horseshoe")
+print("\nWith qualifier expected:")
+print([
+    "For want of a nail the shoe was lost.",
+    "And all for the want of a horseshoe nail."
+])
+print("\nWith qualifier actual:")
+print(result_with_qualifier)
\ No newline at end of file

========== LOGS (198 lines) ==========
```

```
 INFO: [PROBLEM_SUITE] Starting sandbox to evaluate solution diff for problem proverb
DEBUG: [SANDBOX] Created sandbox temp directory for <sandbox_tmp4ooadt9k>: /tmp/tmp4ooadt9k
DEBUG: [POLYGLOT] Copied main.py to /tmp/tmp4ooadt9k/repo for proverb
DEBUG: [POLYGLOT] Copied tests.py to /tmp/tmp4ooadt9k/repo for proverb
DEBUG: [POLYGLOT] Initializing git repository in /tmp/tmp4ooadt9k/repo for proverb
DEBUG: [GIT] Initializing git repository in /tmp/tmp4ooadt9k/repo
DEBUG: [GIT] Initialized git repository in /tmp/tmp4ooadt9k/repo
DEBUG: [GIT] Adding all files in /tmp/tmp4ooadt9k/repo
DEBUG: [GIT] Added all files in /tmp/tmp4ooadt9k/repo
DEBUG: [GIT] Making initial commit: Initial commit
DEBUG: [GIT] Made initial commit: Initial commit
DEBUG: [POLYGLOT] Initialized git repository in /tmp/tmp4ooadt9k/repo for proverb
DEBUG: [PROBLEM_SUITE] Applying agent's solution diff to /tmp/tmp4ooadt9k/repo for problem proverb
DEBUG: [PROBLEM_SUITE] Applied agent's solution diff to /tmp/tmp4ooadt9k/repo for problem proverb
```

```
DEBUG: [SANDBOX] Copied main Python script (/root/abstract-agent-runner/
problem_suites/polyglot/TEST_RUNNER.py) for <sandbox_tmp4ooadt9k>: /tmp/
tmp4ooadt9k/TEST_RUNNER.py
DEBUG: [SANDBOX] Written input.json for <sandbox_tmp4ooadt9k>: /tmp/
tmp4ooadt9k/input.json
DEBUG: [SANDBOX] Running sandbox <sandbox_tmp4ooadt9k>
DEBUG: [SANDBOX] Started sandbox runner thread for <sandbox_tmp4ooadt9k>
DEBUG: [PROBLEM_SUITE] Started sandbox to evaluate solution diff for
problem proverb
DEBUG: [SANDBOX] Cleaned up sandbox <sandbox_tmpk95tjvac>
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Entered
main()
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Loading
main.py
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Loaded
main.py
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Loading
tests.py
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Loaded
tests.py
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Found test
class: ProverbTest
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Found 8 test
methods
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] [1/8] Running
test_an_optional_qualifier_can_be_added...
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER]
test_an_optional_qualifier_can_be_added: PASSED
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] [2/8] Running
test_an_optional_qualifier_in_the_final_consequences...
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER]
test_an_optional_qualifier_in_the_final_consequences: PASSED
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] [3/8] Running
test_four_pieces_modernized...
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER]
test_four_pieces_modernized: PASSED
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] [4/8] Running
test_full_proverb...
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER]
test_full_proverb: PASSED
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] [5/8] Running
test_one_piece...
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER]
test_one_piece: PASSED
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] [6/8] Running
test_three_pieces...
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER]
test_three_pieces: PASSED
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] [7/8] Running
test_two_pieces...
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER]
test_two_pieces: PASSED
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] [8/8] Running
test_zero_pieces...
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER]
test_zero_pieces: PASSED
```

```
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Test results:
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [{'name':
'test_an_optional_qualifier_can_be_added', 'status': 'pass'}, {'name':
'test_an_optional_qualifier_in_the_final_consequences', 'status':
'pass'}, {'name': 'test_four_pieces_modernized', 'status': 'pass'},
{'name': 'test_full_proverb', 'status': 'pass'}, {'name':
'test_one_piece', 'status': 'pass'}, {'name': 'test_three_pieces',
'status': 'pass'}, {'name': 'test_two_pieces', 'status': 'pass'},
{'name': 'test_zero_pieces', 'status': 'pass'}]
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Test summary:
8 passed, 0 failed, 0 skipped
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Writing
output.json
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Wrote
output.json
DEBUG: [DOCKER:sandbox_tmp4ooadt9k] [POLYGLOT_TEST_RUNNER] Exiting
main()
DEBUG: [SANDBOX] <sandbox_tmp4ooadt9k> finished running
DEBUG: [SANDBOX] <sandbox_tmp4ooadt9k> captured 29 lines of logs
DEBUG: [SANDBOX] Read output.json from <sandbox_tmp4ooadt9k>: /tmp/
tmp4ooadt9k/output.json
 INFO: [PROBLEM_SUITE] Finished sandbox to evaluate solution diff for
problem proverb: success



========== TEST RESULTS ==========
8 passed, 0 failed, 0 skipped
test_an_optional_qualifier_can_be_added - no category - pass
test_an_optional_qualifier_in_the_final_consequences - no category -
pass
test_four_pieces_modernized - no category - pass
test_full_proverb - no category - pass
test_one_piece - no category - pass
test_three_pieces - no category - pass
test_two_pieces - no category - pass
test_zero_pieces - no category - pass
========== LOGS (29 lines) ==========



DEBUG: [SANDBOX] Cleaned up sandbox <sandbox_tmp4ooadt9k>
```